Background Subtraction Methods for Online Calibration of Baseline Received Signal Strength in Radio Frequency Sensing Networks

Andrea Edelstein

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

December 2011

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Engineering.

Abstract

Radio frequency (RF) sensing networks are a class of wireless sensor networks (WSNs) which use RF signals to carry out tasks such as tomographic imaging, tomographic target tracking and node localization. While a wide variety of such algorithms exist, they often assume access to measurements of the baseline received signal strength (RSS) on each link, i.e, to measurements taken during some offline calibration period when no temporary obstructions are located near the nodes which form the network. However, in many cases, WSNs are designed to be deployed and used on the fly, and it can be impossible to ensure the network is empty of obstructions long enough to obtain the required calibration data. For instance, an RF sensing network could be set up around a burning building to image its interior and determine if people are trapped inside. There is no way to ask these people to first leave the area while the baseline RSS values are collected.

Thus far, no research has addressed the question of whether it is possible to estimate baseline RSS values without access to a calibration period. We propose adapting background subtraction methods from the fields of computer vision and image processing in order to estimate baseline RSS values from measurements taken while the system is online and obstructions may be present in the network. This is done by forming an analogy between the intensity of a background pixel in an image and the baseline RSS value of a link in a WSN. We also translate the concepts of temporal similarity, spatial similarity and spatial ergodicity which underlie three specific background subtraction algorithms—background subtraction with temporal background modelling, foreground-adaptive background subtraction and foreground-adaptive background subtraction with Markov modelling of change labels—to the domain of WSNs in order to use these algorithms to determine the baseline RSS.

By applying these techniques to experimental data, we show that they are capable of accurately estimating baseline RSS values in a range of different environments. We also show that these estimates are close enough to the actual values of the baseline RSS to allow for RF tomographic tracking to be carried out without the need to resort to a calibration period.

Abrégé

Les réseaux de capteurs à radiofréquences sont une classe de réseau de capteurs sans fil qui utilisent des signaux radioélectriques pour accomplir de nombreuses tâches comme l'imagerie tomographique, la poursuite tomographique de cibles, la localisation des noeuds, etc. Même s'il existe une grande variété de ces algorithmes, la plupart assument que des niveaux de référence pour la force du signal entre deux noeuds peuvent être obtenus pendant une période d'étalonnage différée, c'est-à-dire quand il n'y a aucun obstacle temporaire près des noeuds du réseau. Toutefois, les réseaux de capteurs sans fil sont conçus pour être déployés et utilisés de façon ad hoc, ce qui rend parfois impossible l'évacuation des obstacles pour que ces niveaux de référence puissent être mesurés. Par exemple, un réseau de capteurs à radiofréquence peut être installé autour d'un bâtiment en feu, et l'imagerie tomographique peut être utilisée pour déterminer si quelqu'un est piégé à l'intérieur. Évidemment, il est irréaliste de demander aux personnes de quitter la région en premier lieu pendant qu'on établit les niveaux de référence.

Jusqu'à présent, la recherche existante ne s'est pas penchée sur la possibilité d'estimer ces niveaux de référence sans période d'étalonnage. Nous proposons d'adapter la méthode de soustraction de l'arrière-plan — algorithme créé à l'origine pour la vision artificielle et pour le traitement des images — pour estimer les niveaux de référence pour la force du signal en utilisant des mesures prises quand le système est en ligne et quand on retrouve possiblement des obstacles dans les environs du réseau. Cette adaptation consiste, entre autres, à former une relation entre l'intensité d'un pixel arrière-plan et le niveau de référence de la force du signal sur une liaison d'un réseau de capteurs sans fil. Nous adaptons aussi les concepts de la similarité temporelle, de la similarité spatiale et de l'ergodicité spatiale qui sous-tendent trois méthodes de soustraction de l'arrière-plan en vue de les utiliser pour trouver les niveaux de référence de la force du signal.

Avec ces techniques, nous montrons que nous sommes capables d'estimer les niveaux de référence de la force du signal dans plusieurs environnements différents. Nous montrons aussi que ces estimations sont assez précises pour que la poursuite tomographique de cible puisse être effectuée sans avoir besoin d'une période d'étalonnage.

Acknowledgments

I must start by thanking my supervisor, Prof. Michael Rabbat, for the immense help he has provided over the course of this project. I greatly appreciate his patience, advice and the sheer amount of time he has dedicated to me and to his other students. I would also like to thank Prof. Mark Coates for his insights into the domain of RF tomography.

I must acknowledge, here, our collaborators in China at the Beijing University of Posts and Telecommunication, specifically Wen Chin, Xiaobo Chen and Yunpeng Li, who dealt promptly and graciously with my many questions. Thanks go to them.

Thanks must also go to every member of McGill University's Computer Networks Research Lab, all of whom contributed to this work in some way, whether by participating in experiments, by helping to troubleshoot code or explain concepts or simply by providing moral support. I would like to thank Abhay Ghatpande for overseeing my preliminary research, and I must single out Xi Chen in particular for his tireless work on the RF tomography project over the last two years.

I would also like to thank my parents for their love, their constant support...and their invaluable lifts to and from McGill.

Finally, I must acknowledge a grant received from the Canadian Natural Sciences and Engineering Research Council (NSERC) which helped support this research.

Contents

List of Figures

List of Tables

List of Acronyms

RF	Radio Frequency
WSN	Wireless Sensor Network
RSS	Received Signal Strength
LOS	Line-Of-Sight
NLOS	Non-Line-Of-Sight
PDF	Probability Density Function
TBM	Temporal Background Modelling
FABS	Foreground-Adaptive Background Subtraction
FABS-MMCL	FABS with Markov Modelling of Change Labels
FABS-MMCL-R	Rectangle-based FABS-MMCL
FABS-MMCL-O	Obstruction-based FABS-MMCL
MA	Mean Approximation
RMSE	Root Mean Square Error

Chapter 1

Introduction

1.1 Overview

Radio frequency (RF) tomography is a promising field of research which uses information gleaned from the strength of received wireless signals in order to construct images of physical environments. RF tomography can be used to image static environments, but many of the technology's most interesting applications are also designed to investigate specific aspects of how an environment changes over time. For instance, RF tomographic tracking can be used to track the movements of a walking person, through walls, without that person needing to carry a GPS (global positioning system) unit, a transponder or any other type of special equipment.

In order for RF tomography algorithms to function properly, they require access to calibration values: a set of wireless signal strength measurements taken over a period of time when the system is deployed and transmitting packets, but not yet officially online. During this time, the environment must be static and empty of temporary obstructions (i.e., in a "background" state). Some RF tomography algorithms then make use of these measurements directly; other algorithms—designed for use in time-varying environments—go on to compare them with measurements taken later on, discerning information about how the environment is changing from how these measurements differ from the calibration values. Unfortunately, these calibration values are often difficult to obtain, since it may be beyond our power to force the environment to remain static and empty long enough for them to be collected.

In this thesis, we propose adapting a technique from the fields of computer vision and image processing—namely, background subtraction—in order to obtain an estimate for these calibration values using measurements taken once the system is already online, i.e., when no calibration period is available.

1.2 Motivation

The wireless signal measurements required for RF tomography are typically collected using wireless sensor networks (WSNs). These networks consist of many small pieces of hardware, known as nodes or motes, which are distributed around some region of interest. Each of the nodes is traditionally equipped with one or more devices which can sense some physical feature of the surrounding environment [1]. These sensors may range from the simple—such as thermometers to measure ambient temperature—to the complex—such as devices which determine water quality in lakes and rivers by measuring pH levels, conductivity, etc. [2]. Crucially, WSN nodes are also equipped with wireless transceivers so that they can pass along the measurements they take, either amongst themselves or else to some central authority.

Because of the ubiquity of these transceivers on wireless nodes, RF tomography applications were developed to treat the signals of the transceivers *themselves* as the data to be sensed [3]. In other words, RF tomography applications recover important information not from external temperature or pH sensors, but rather by using the nodes' own hardware to measure the received signal strength of the messages sent along the wireless links stretching between the nodes.

Received signal strength (RSS) refers to the power carried by a transmission, expressed in dBm (decibels above a reference level of 1 mw), measured at the receiving node. The RSS of a signal depends on—among other factors—the transmitting power of the sender from which it originates, the distance between the sender and the receiver, and the layout of the environment surrounding the sender and the receiver (especially the presence and physical composition of any obstructing objects which may be present). Although these relationships can be complicated, RF tomography still manages to take advantage of them. In a static environment, knowledge of the physical laws which govern how obstructions attenuate and affect RSS measurements is exploited by RF tomographic imaging methods to build three-dimensional representations of the objects or landscape in the region of interest surrounding the nodes, in a process where radio waves are used analogously to the X-rays employed in computed tomography (CT) scans [4, 5]; in a non-static environment, by comparing how RSS measurements on different links change over time, RF tomographic tracking can be used to follow the movements of people or objects in the region of interest [6–11]. A class of algorithms known as device-free passive localization techniques can determine the location of an obstruction by comparing the current values of the RSS on each link to those which have previously been recorded in a passive radio map, a data structure which stores information about the values the RSS was seen to take in the past when an obstruction was known to be present in certain locations [12, 13]. Although not specifically under the umbrella of RF tomography, RSS-based node localization also employs RSS in order to determine nodes' relative positions, converting RSS measurements into measurements of distance [14, 15].

It is worth noting that problems such as imaging, tracking and localization, which can be solved by inspecting RSS measurements, can also be addressed via other methods as well: the whole issue of node localization can be avoided if the locations of the nodes are simply recorded as they are deployed; tracking can be carried out precisely and accurately by providing the objects being tracked with RFID (radio frequency ID) cards which can then be monitored using specialized readers [16]. Where WSNs which analyze RSS measurements ("RF sensing networks") fill a gap is in applications where we do not have an opportunity, before-hand, to properly initialize our system, e.g., when we do not have time to record the locations of our nodes or when we do not have the foreknowledge to issue RFID cards to the people who will wander into the vicinity of the network. In this vein, Wilson and Patwari [7] describe a scenario where nodes are "launched" around a building that is too dangerous to approach (perhaps because of a fire), in order to track and locate people who may be inside it. Similarly, Patwari and Wilson [17] also mention that if police are aware that an unauthorized intruder has entered a building, they can quickly scatter nodes around its perimeter in order to determine the intruder's exact location without putting themselves in harm's way by manually searching the building. These are perfect illustrations of the benefits of RF sensing networks. Because of the small size and relatively low cost of the nodes needed to construct these networks, they are uniquely suited to be deployed quickly and easily to solve problems in all sorts of environments.

Unfortunately, the algorithms which have been developed for RF sensing networks cannot actually work in their intended fashion because they require access to calibration data which must be collected before the system is brought online, during a period of time when we can be assured that no temporary obstructions are present in the network. For instance, before the tracking and imaging algorithms described in [4, 7, 8] can function properly, they must first calibrate the mean RSS value on each link in the network by collecting measurements during a window of time where no moving targets are present in the region of interest. In the case of node localization or tomographic imaging of landscapes, to avoid incorporating the effects of transitory objects such as people or animals into the estimated positions of the nodes or into the tomographic image being built, it is best to work with measurements collected at a time when such objects are not present. In all cases, it may be difficult or impossible to evacuate the region of interest for a long enough period of time to obtain the required calibration measurements.

1.3 Problem Description

In order for RF sensing networks to achieve their true potential to be used on the fly, we would like to find a way to approximate these required "baseline" RSS values, seen on each link in the network when no temporary obstructions are present to affect said values.

A simple way to approximate the baseline RSS value for a given link might be to average many measurements for that link over time, with some of these measurements doubtlessly being taken while the RSS value is perturbed by the presence of an obstruction but (hopefully) many more measurements being taken while the RSS is closer to its baseline value. Over time, we might hope that the latter values would overwhelm the former, but this is certainly not guaranteed, depending on how long the obstruction lingers in a certain place or how its path aligns with the positions of the nodes and links in the network.

Because of these issues, it would be even better if we could approximate the baseline RSS for a given link by averaging *only* over measurements taken on that link at times when we know that it is not obstructed. In order to do this—even in cases where we do not have access to any information about the locations of the obstructions—we propose to adapt background subtraction methods from the fields of computer vision and image processing and apply these methods to the links in WSNs. Background subtraction algorithms are designed to determine which pixels in a frame of video image are currently static, background pixels [18]. We use the same algorithms to determine which links in a WSN are currently unaffected by obstructions and hence in the "background" and which links are currently affected and hence in the "foreground." By then taking the mean of the measurements seen on a link only at times when that link is in the background, we can achieve a better estimate of the link's baseline RSS than can be found by merely taking the mean over all available measurements, all without the need for an offline calibration period during which time the network must be held empty.

1.4 Thesis Contribution

Different background subtraction algorithms currently exist to exploit different features of videos and images. For instance, concepts such as the temporal similarity of video and the spatial similarity and ergodicity of still frames can be taken into consideration in order to improve background subtraction performance. In this thesis, we explore how to translate these concepts to the links of WSNs in order to apply background subtraction to this new domain. We then use experimental data to evaluate the performance of this novel online baseline RSS estimation technique. A paper based on this work has been submitted for consideration to the IEEE Transactions on Mobile Computing.

1.5 Thesis Organization

Chapter 2 explains how RF sensing networks are usually deployed and mentions some of the difficulties they can encounter when interpreting RSS values. The chapter then presents several RF sensing network applications—namely RF tomographic imaging, mean-based RF tomographic tracking, variance-based RF tomographic tracking and RSS-based node localization—in more detail in order to further establish the importance of estimating baseline RSS.

Chapter 3 gives a general description of background subtraction as it is currently used in the fields of computer vision and image processing. Chapter 3 then also provides an in-depth explanation of three particular background subtraction algorithms which we select to apply to WSNs.

Chapter 4 begins by explaining why background subtraction is well-suited for use in WSNs. This chapter then explains the modifications which need to be made to the aforementioned algorithms in order to adapt them for use with wireless links instead of with images and pixels, concentrating on explaining how to translate the concepts of temporal similarity, spatial similarity and spatial ergodicity.

Chapter 5 presents a comparison of the performance of these background subtraction algorithms, using experimental data collected at the Beijing University of Posts and Telecommunications in Beijing, China.

Chapter 6 summarizes our work and discusses areas for future research.

Chapter 2

Overview of RF Sensing Networks

As previously mentioned, several different RF sensing network applications currently exist. Section 2.1 presents an overview of how these networks are usually constructed. Section 2.2 briefly explains some of the unintuitive behaviours which can be encountered when working with RSS measurements in such networks. Finally, Sections 2.3, 2.4, 2.5 and 2.6 talk about four particular applications of RF sensing networks: tomographic imaging, mean- and variance-based tomographic tracking and node localization, all of which require access to baseline RSS values in order to function.

2.1 RF Sensing Network Construction and Operation

Like WSNs in general, RF sensing networks are constructed from a series of nodes which can be carefully placed or carelessly scattered around a region of interest. The nodes in the network are programmed to communicate amongst themselves by sending each other packets via wireless links. Depending upon the placement of the nodes and the desires of the network developer, all nodes may communicate with all other nodes (forming a complete graph) or some other network topology may be preferred. If the nodes are located far enough apart from one another, the possible network topologies may be limited by the transmission ranges of the nodes.

In addition to different network topologies, WSNs can also be built using different transmission protocols. A simple token-ring protocol is often implemented whereby each node transmits during a specified time slot, with only a single node transmitting in each slot, but other protocols can be employed as well. Packets may be sent directly to a specified recipient or broadcast to all other nodes. In any case, the packets sent between the nodes usually contain identifying information about the sender (e.g., a node ID) along with additional information (e.g., data to help administer the token ring). Receiving nodes interpret this information and, crucially, measure the RSS of the incoming packet. In this way, the RSS value of the sender-receiver link can be determined. This information can then be processed by the sensor nodes or transmitted to a central authority for further analysis.

For the RF sensing applications relevant to this work, such as tomographic imaging (Section 2.3), tomographic tracking (Sections 2.4 and 2.5) and node localization (Section 2.6), the most desirable network configuration is one which allows the RSS on every link in the network to be probed simultaneously. However, because WSNs are subject to interference and packet collisions when multiple nodes transmit at the same time, this is almost never realistic. In practice, RSS measurements are usually obtained by relying on the aforementioned token-ring protocol as follows: for a network with Φ nodes, node 1 broadcasts a packet to all the other nodes in the network. This establishes measurements for the RSS on $\Phi - 1$ unidirectional links which stretch from node 1 to node 2, from node 1 to node 3, etc. Next, node 2 broadcasts a packet to all the other nodes on the network, yielding another series of $\Phi - 1$ unidirectional RSS measurements. This continues until all Φ nodes have broadcast, at which point node 1 broadcasts again [9]. Many WSN applications then implicitly bundle together a small number of these consecutive $(\Phi - 1)$-sized measurement vectors (which have been collected over a short period of time) and go on to treat them as though they were all collected simultaneously, i.e., as though these measurement vectors, taken together, can provide an instantaneous capture of the RSS values on most or all of the links in the network at a single instant. This is only possible under the assumption that the conditions in the network remain static in the time it takes to collect these multiple measurement vectors, but since WSNs are usually capable of sending upwards of several dozens of packets per second, this assumption is sound.

2.2 Received Signal Strength Theory

As previously mentioned, once the RSS measurements are collected, they can be processed either centrally or in a distributed fashion. In either case, for RF sensing networks, this analysis depends heavily on pre-existing models which explain how signal strength is affected by distance, by solid obstructions, etc. In general, the vector of RSS measurements $\mathbf{R}^{(k)}$ (measured in dBm over all the links in the network over a short period of time denoted by k) can be thought of as

$$\mathbf{R}^{(k)} = \mathbf{R_B} - \mathbf{R_F}^{(k)} \tag{2.1}$$

where $\mathbf{R_B}$ represents the baseline, "background" components of the RSS signal, which are static over the short term (although they may change on timescales of hours or days), and $\mathbf{R_F}^{(k)}$ represents its time-varying, "foreground" components. Note that only $\mathbf{R}^{(k)}$ can be measured directly, yet most practical applications of RF sensing networks require access to $\mathbf{R_B}$ and/or $\mathbf{R_F}^{(k)}$.

If the network is empty of temporary obstructions, $\mathbf{R_F}^{(k)}$ should simply equal $\mathbf{0}$. On the other hand, the value of $\mathbf{R_B}$ can be influenced by a multitude of factors. Wilson and Patwari [4] express $\mathbf{R_B}$ as

$$\mathbf{R_B} = \mathbf{P_s} - \mathbf{L_d} - \mathbf{L_a} - \mathbf{S_o} \tag{2.2}$$

where $\mathbf{P_s}$ is the transmitted power emanating from the sender; $\mathbf{L_d}$ is the large-scale path loss due to distance d; $\mathbf{L_a}$ represents the static loss due to antenna patterns, device inconsistencies, etc. and $\mathbf{S_o}$ is the shadowing loss due to static objects.

The transmitted power is generally known to the developers of the WSN and the other terms in Equation (2.2) can be modelled in a variety of ways. For instance, the log-distance path loss model [19] is often used to calculate $\mathbf{L_d}$ for a vector of links with lengths $\mathbf{d}$. This model states that RSS decreases exponentially with respect to distance. More specifically, it says that

$$\mathbf{L_d} = L_{d_0} + 10 n_p \log\left(\frac{\mathbf{d}}{d_0}\right) + \mathbf{w}, \tag{2.3}$$

where L_{d_0} is the path loss at a reference distance d_0 from the transmitter (usually 1 m), n_p is the path loss coefficient (typically between 2 and 4) which varies depending on the medium the signal is travelling through and $\mathbf{w}$ represents white Gaussian noise. The path loss at the reference distance is often estimated using the Friis free space equation

$$L_{d_0} = -10 \log\left(\frac{G_s G_r \lambda^2}{16\pi^2 d_0^2}\right) \tag{2.4}$$

where G_s and G_r are the antenna gains of the sender and the receiver respectively and λ is the wavelength of the transmitted signal [19].

The noise in Equation (2.3) can often be quite significant (in our experiments, discussed in greater detail in Chapter 5, the variance of $\mathbf{w}$ on unobstructed links sometimes reached 10 dBm^2 or more) and represents a major hurdle to RF sensing networks. Furthermore, research has shown that the variance of this noise will increase on links which are obstructed by moving objects [7].

In addition to affecting the variance of the RSS, obstructions will also affect the RSS values themselves. Indeed, it seems fairly intuitive to assume that if an obstruction is located on the line-of-sight (LOS) path between two nodes, the wireless signals exchanged by these nodes will always experience a decrease in RSS. This, however, is not necessarily the case, largely due to the non-line-of-sight (NLOS) components of wireless signals: while a good deal of the energy carried by these signals will tend to travel along the LOS path between sender and receiver, a not-inconsiderable portion also travels along various indirect NLOS paths [17]. This has two main implications. First, the RSS value of a link may be affected by obstructions which are not located along the LOS path between its sender and receiver (and, conversely, this RSS may not be significantly affected by obstructions which *are* on this path). Second, it is well-documented that, by causing phase differences to manifest between the LOS and NLOS components of the signal, obstructions can also cause constructive interference, actually leading to *increases* in the RSS on the link they obstruct [20]. These "multipath" effects are less intense—though still present—in wide-open environments such as extremely large rooms or outdoor fields, where signals do not have as many opportunities to reflect off of walls and furniture to create multiple NLOS paths. On the other hand, in regular indoor environments such as homes or offices, multipath effects represent another significant stumbling block to the efficacy of RF sensing networks.

Even in environments where multipath effects are negligible, the effect of obstructions on RSS is still complex, particularly for cases where multiple obstructions are present between the sender and the receiver. Many models exist to quantify the effect of a small number of obstructions, but the effect of large numbers of obstructions is much harder to predict.

For a single obstruction, one commonly-used technique, the knife-edge diffraction model [19], states that a signal (sent by a node ν_s), which would normally have magnitude $|E|$ at a receiver ν_r, will have a magnitude $|E'|$ in the presence of an obstruction, where

$$|E'| = |E| \left(\frac{S(v) + 0.5}{\sqrt{2} \sin\left(\Delta_\phi + \frac{\pi}{4}\right)} \right), \tag{2.5}$$

$$\Delta_\phi = \tan^{-1} \left(\frac{S(v) + 0.5}{C(v) + 0.5} \right) - \frac{\pi}{4}, \tag{2.6}$$

$$v = -h_p \sqrt{\frac{2}{\lambda} \left(\frac{1}{d_s} + \frac{1}{d_r} \right)}, \tag{2.7}$$

where $C(v)$ and $S(v)$ are Fresnel integrals and where h_p, d_s and d_r represent the physical relationship between the sender, the receiver and the obstruction as seen in Figure 2.1.

The knife-edge diffraction model illuminates the extremely complicated relationship which exists between obstructions and RSS. While other models exist which define this effect differently, any RF sensing network application which hopes to deal properly with obstructions must nonetheless take the immense complexity of this relationship into account.

Finally, even if no obstructions are present in the region of interest, i.e., even if $\mathbf{R_F}^{(k)} = \mathbf{0}$ for all values of k, the RSS measured between a sender and a receiver may still change over longer timescales, e.g., as the nodes' batteries wear down [21] or if the nodes' relative positions or orientations shift slightly due to vibrations or wind.

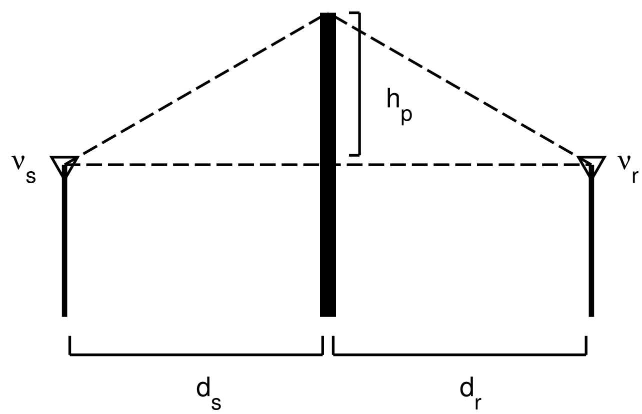


Fig. 2.1 An illustration of the relationship between the transmitter ν_s, the receiver ν_r and a solid obstruction (represented by the thick black line), according to the knife-edge diffraction model. The quantities h_p, d_s and d_r are indicated in the figure [19].

2.3 RF Tomographic Imaging

Despite all these complications, we can still say that every link ℓ in a WSN has some relatively static baseline RSS value $R_B[\ell]$ which depends on the physical and operational properties of the nodes forming the link, their physical separation from one another and the composition of the environment surrounding them. If the RSS measurement $R^{(k)}[\ell]$ seen on ℓ deviates greatly from this baseline RSS at any time, this probably indicates that one of these factors has changed in some way. If we assume that the factors which contribute to $R_B[\ell]$ such as the nodes' positions, antenna properties, etc. have not been altered, it then becomes probable that this deviation is attributable to the effects of $R_F^{(k)}[\ell]$, most likely in the form of the presence of some obstruction in the region of interest. RF tomographic imaging takes advantage of this probability by inspecting such deviations in order to reconstruct the layouts of these obstructions.

For instance, in Wilson and Patwari's RF tomographic imaging algorithm described in [7], an array of nodes is placed around a region of interest inside of which objects will eventually be moving. After the nodes are set in place, they are allowed to transmit amongst themselves for several minutes while no moving obstructions are present in the network (i.e., while $\mathbf{R_F}^{(k)} = \mathbf{0}$) in order to estimate $\mathbf{R_B}$. After this baseline RSS has been measured, obstructions are allowed to enter the region of interest, while the nodes continue to probe the links' RSS values. The measured value of $\mathbf{R_B}$ is then subtracted from $\mathbf{R}^{(k)}$ in order to determine the entries of $\mathbf{R_F}^{(k)}$.

For simplicity's sake, Wilson and Patwari assume that, for a given link, any non-zero value of $R_F^{(k)}[\ell]$ is attributable to the presence of an obstruction located somewhere along the direct LOS path of link ℓ; more specifically, by quantizing the region of interest into a set of small cubic voxels, they assume that $R_F^{(k)}[\ell]$ is a weighted sum of the attenuation seen in every voxel through which ℓ passes. RF tomographic imaging is then the process of determining the attenuation $x_j^{(k)}$ in voxel j during the period of time k which would cause the observed value of $\mathbf{R_F}^{(k)}$ over all links. When all values of $x^{(k)}$ are collected into the vector $\mathbf{x}^{(k)}$, this vector directly represents an image of the obstruction field present in the region of interest. A graphical representation of this process can be seen in Figure 2.2.

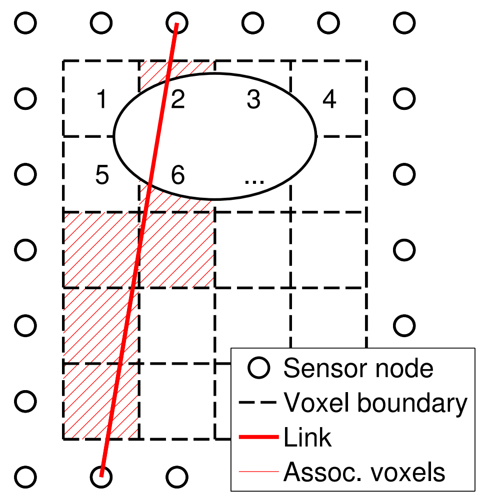


Fig. 2.2 A graphical representation of the theory underlying RF tomographic imaging as described in the work of Wilson and Patwari [4]. The value of $\mathbf{R_F}^{(k)}$ on the indicated link can be thought of as the sum of the attenuation $x_j^{(k)}$ in all of the shaded voxels. In this case, this attenuation is only non-zero in voxels 1–8 which are occupied by the solid obstruction. Therefore, the value of $\mathbf{R_F}^{(k)}$ is the sum of the attenuation in voxels 2 and 6.

This formulation leads to a simple equation whereby $\mathbf{R_F}^{(k)}$ is equal to

$$\mathbf{R_F}^{(k)} = \mathbf{W}\mathbf{x}^{(k)} + \mathbf{g}_\mu \tag{2.8}$$

where $\mathbf{W}$ is a weighting matrix which encodes which links pass through which voxels and $\mathbf{g}_\mu$ is a vector of i.i.d. Gaussian noise with variance σ_μ^2.

Wilson and Patwari [4] use a so-called elliptical model for $W_{\ell j}$ (the value of $\mathbf{W}$ for link ℓ and voxel j) such that

$$W_{\ell j} = \frac{1}{\sqrt{d_\ell}} \begin{cases} 1 & \text{if } d_{sj} + d_{rj} < d_\ell + \kappa \\ 0 & \text{otherwise} \end{cases} \tag{2.9}$$

where d_ℓ is the length of link ℓ, d_{sj} and d_{rj} are the distances between the sending node of link ℓ and the center of voxel j and between the receiving node of link ℓ and the center of voxel j respectively, and κ is a tuneable parameter which expresses the "width" of the LOS path. Of course, this formulation is obviously a simplification since, due to multipath effects, every voxel in the network can potentially contribute to every entry in $\mathbf{R_F}^{(k)}$.

Wilson and Patwari also assume that $\mathbf{x}^{(k)}$ is a zero-mean Gaussian random field with an a priori covariance matrix $\mathbf{C}_x$ and then use a maximum a posteriori formulation to solve Equation (2.8) for $\mathbf{x}^{(k)}$ such that

$$\hat{\mathbf{x}}_{MAP}^{(k)} = \arg\max_{\mathbf{x}^{(k)}} \ln P\left(\mathbf{x}^{(k)} | \mathbf{R_F}^{(k)}\right) \tag{2.10}$$

$$= \left(\mathbf{W}^T\mathbf{W} + \mathbf{C}_x^{-1}\sigma_\mu^2\right)^{-1} \mathbf{W}^T \mathbf{R_F}^{(k)}. \tag{2.11}$$

Although this algorithm contains a number of simplifying assumptions, especially with respect to the composition of $\mathbf{W}$, Wilson and Patwari have successfully managed to implement RF tomographic imaging in a variety of scenarios, provided the WSN network always has access to a calibration period in which to measure the baseline RSS. Without this calibration period, though—and hence with no way to know the value of $\mathbf{R_B}$—their algorithm does not specify how to isolate the effects of $\mathbf{R_F}^{(k)}$ from the measured values of $\mathbf{R}^{(k)}$.

2.4 Mean-Based RF Tomographic Tracking

The mean-based RF tomographic tracking algorithm described by Li et al. and Chen [8,9] operates on a similar principle to RF tomographic imaging. Again, an array of nodes is deployed around the region of interest and left to measure the baseline RSS on each link; only once this is done are obstructions permitted to enter the region. Again, too, the baseline RSS is subtracted from the measurements taken during this latter phase. However, here, the change in the RSS is fed into a particle filter which—given access to these changes, to knowledge of the prior position of the obstruction and to a model of its movement patterns—can quite accurately track the obstruction's movements.

Particle filters (also known as sequential Monte Carlo methods) are designed to use observable data to estimate hidden variables modelled by Markov chains. In this case, a particle filter is used to estimate $\mathbf{z}_t \in \mathbb{R}^2$, the position of the target obstruction at time t, from $\mathbf{R}_{\mathbf{F}_{1:t}}$, a matrix of the changes observed on all links between the baseline RSS and current RSS measurements, from time 1 to t. (Note that for simplicity, in this section we refer to $\mathbf{R}_{\mathbf{F}_t}$ instead of to $\mathbf{R}_{\mathbf{F}}{}^{(k)}$ to refer to an idealized situation where measurements are taken simultaneously on each link.) In Li et al. and Chen's algorithm, the particle filter is initialized by drawing 1000 2-D vector particles from a uniform distribution, with each particle representing a possible value for $\mathbf{z}_1$. The weights of each particle are then resampled based on the ability of the particle to explain the observed value of $\mathbf{R}_{\mathbf{F}_1}$.

The relationship between $\mathbf{R}_{\mathbf{F}t}$ and $\mathbf{z}_t$ is encoded in a model which states that

$$\mathbf{R}_{\mathbf{F}_t} = \phi \times \mathbf{\Gamma}_t + \mathbf{g} \tag{2.12}$$

where the parameter ϕ represents the mean value of the attenuation we expect the target to cause, and $\mathbf{g}$ represents white Gaussian noise. Furthermore, $\mathbf{\Gamma}_t$ is a matrix with an entry $\Gamma_{t\ell}$ for each link ℓ, defined as

$$\Gamma_{t\ell} = \exp\left(-\frac{d_{s\mathbf{z}_t} + d_{r\mathbf{z}_t} - d_\ell}{2\sigma_\kappa}\right). \tag{2.13}$$

In this model, $d_{s\mathbf{z}_t}$ and $d_{r\mathbf{z}_t}$ are the distances between $\mathbf{z}_t$ and the sender of link ℓ and between $\mathbf{z}_t$ and the receiver of link ℓ respectively. Recall that d_ℓ is the length of the link ℓ. Finally, σ_κ is related to the κ from RF tomographic imaging in that it models how quickly this attenuation decays as distance from the target increases.

Once the particle weights have been resampled and normalized to sum to 1, a weighted average of the particle locations can be taken in order to determine a new approximation for $\mathbf{z}_2$. The particles are then replaced with random samples drawn from an importance function $q\left(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{R}_{\mathbf{F}_t}\right)$ and the process is repeated for the next time step.

Simultaneously, Li et al. and Chen also use an expectation-maximization technique to estimate ϕ, σ_κ and the variance of $\mathbf{g}$ in order to improve tracking performance.

Ideally, as the estimates of ϕ, σ_κ and the noise variance converge to these parameters' true values, and as the importance function is progressively refined in each step of the algorithm, the weighted average of the particle locations should converge to the true location of the obstruction, resulting in it being tracked successfully. In some cases, depending on the quality of the obtained RSS measurements and the exact realizations obtained from the various random distributions, it is possible for the parameter estimates and/or the particle weights to fail to converge, resulting in the particle filter failing to track the obstruction. Encouragingly, in the vast majority of experiments conducted in outdoor and simple indoor environments, Li et al. and Chen's RF tomographic tracking functions admirably. Again, however, without access to a calibration period in which to determine $\mathbf{R_B}$, no concrete suggestions are made in the work of Li et al. and Chen as to how to determine the values of ${\mathbf{R_F}}^{(k)}$ which underlie the algorithm.

2.5 Variance-Based RF Tomographic Tracking

Li et al. and Chen's tracking algorithm [8, 9] relies on the actual value of $\mathbf{R}^{(k)}$—combined with knowledge of $\mathbf{R_B}$—in order to perform tracking. Algorithms such as the one proposed by Wilson and Patwari [7] also exist which can track a target based on the variance of $\mathbf{R}^{(k)}$.

As mentioned in Section 2.2, links which are obstructed tend to experience higher variance. Variance-based RF tomographic tracking algorithms take advantage of this knowledge. These types of algorithms do not actually require access to the mean values of $\mathbf{R_B}$ and ${\mathbf{R_F}}^{(k)}$; rather their performance can be improved if they have access to calibration data for the baseline value of the *variance* of the RSS on each link. As there are parallels between the problem of determining this value and the problem of determining $\mathbf{R_B}$, we include a brief overview of variance-based RF tomographic tracking.

Similarly to the RF tomographic imaging algorithm described in Section 2.3, when carrying out variance-based RF tomographic tracking, Wilson and Patwari again divide the region of interest into voxels and set up a relationship

$$\mathbf{s} = \mathbf{W}'\mathbf{u} + \mathbf{g}_\sigma \tag{2.14}$$

which resembles the one seen in Equation (2.8). Here, however, $\mathbf{s}$ is a vector of the variance of the RSS on each link and $\mathbf{u}$ is a binary representation of the movement (as opposed to the attenuation) in each voxel (i.e., its entries are set to 1 for voxels where motion occurs and 0 otherwise), with $\mathbf{s}$ and $\mathbf{u}$ incorporating measurements and events respectively from across several time periods k. The matrix $\mathbf{W}'$ is merely the weighting matrix seen in Equation (2.9) multiplied by a normalization factor, and $\mathbf{g}_\sigma$ is a vector of i.i.d. Gaussian noise.

Wilson and Patwari solve Equation (2.14) for $\mathbf{u}$ using Tikhonov regularization such that

$$\mathbf{u}_{Tik} = \left(\mathbf{W}'^T\mathbf{W}' + \zeta\mathbf{Q}^T\mathbf{Q}\right)^{-1}\mathbf{W}'^T\mathbf{s} \tag{2.15}$$

where $\mathbf{Q}$ is a Tikhonov matrix and ζ is a tuneable regularization parameter.

The maximum value of $\mathbf{u}_{Tik}$ is then used as an input to a Kalman filter which, like the particle filter described in Section 2.4, can track a target's motion using this input combined with knowledge of the measurement noise, the target's past locations and the target's presumed movement patterns.

Note that the baseline variance of the RSS is not explicitly used anywhere in this formulation. This is because Wilson and Patwari assume that it is equal to $\mathbf{0}$. As we mention in Section 2.2, however, this is definitely not the case. Therefore, the performance of variance-based RF tomographic tracking algorithms can be improved with access to the actual values of the baseline variance.

2.6 RSS-Based Node Localization

In addition to requiring access to the baseline RSS values for each link, both tomographic imaging and tomographic tracking algorithms require knowledge of the relative positions of the nodes in the network. For instance, for the aforementioned algorithms, this information is required in order to construct $\mathbf{W}$, $\mathbf{W}'$ and $\mathbf{\Gamma}_t$.

These node positions are often simply recorded when the nodes are first deployed, but if they are not, they must be determined in some other way. Wilson and Patwari [7] posit that nodes could be outfitted with GPS units for this purpose while Chen et al. [10] allow for the possibility that the nodes could use an RSS-based node localization algorithm to self-localize. In this latter case, we once more run into the issue of needing access to the baseline RSS. This is because RSS-based node localization operates by converting RSS measurements into distance measurements, often by following a simple formula such as the log-distance path loss model seen in Equation (2.3). Although this may not be the case, RSS-based node localization algorithms assume that $\mathbf{R_B}$ is dominated by the effects of $\mathbf{P_s}$ and $\mathbf{L_d}$—the sender's power and the path loss on every link—and that ${\mathbf{R_F}}^{(k)} = \mathbf{0}$ in order to be able to say that

$$\begin{aligned} \mathbf{R} = \mathbf{R_B} &= \mathbf{P_s} - \mathbf{L_d} && (2.16) \\ &= \mathbf{P_s} - \left(L_{d_0} - 10 n_p \log\left(\frac{\mathbf{d}}{d_0}\right) + \mathbf{w}\right). && (2.17) \end{aligned}$$

Given L_{d_0}, d_0, n_p, $\mathbf{P_s}$ and $\mathbf{R}$ (from measurements), node localization solves this equation for $\mathbf{d}$ and then uses a variety of minimization techniques to find a coordinate vector for the nodes which best satisfies the overdetermined system specified by the internode distances. However, if $\mathbf{R_B} \neq \mathbf{R}$, i.e., if $\mathbf{R}^{(k)} = \mathbf{R_B} + {\mathbf{R_F}}^{(k)}$ is used on the left-hand side of Equation (2.16), the effect of temporary obstructions in the region of interest will become erroneously incorporated into the estimate for $\mathbf{d}$. Therefore, in order to obtain a more accurate estimate for $\mathbf{d}$, it is important that the node localization algorithm be used with measurements of $\mathbf{R_B}$—which heretofore have only been available during the calibration period.

We have now discussed some of the difficulties inherent in using RSS measurements in practical applications. We have also described a variety of algorithms and applications which nonetheless manage to use RSS to image environments, track moving obstructions and determine the positions of wireless sensor nodes. We now leave RF sensing networks for the time being in order to describe the technique of background subtraction.

Chapter 3

Overview of Background Subtraction

Section 3.1 gives a brief overview of background subtraction as it was originally intended to be applied to still frames captured from videos. Then, Sections 3.2, 3.3 and 3.4 present the details of three specific background subtraction algorithms: namely, background subtraction with temporal background modelling, foreground-adaptive background subtraction and foreground-adaptive background subtraction with Markov modelling of change labels. These constitute the background subtraction methods which will be adapted and applied to the previously-described RF sensing networks in future chapters.

3.1 Traditional Background Subtraction

Background subtraction techniques originated in the fields of computer vision and image processing. Using video taken from a stationary camera, the goal of background subtraction is the detection and differentiation of any moving "foreground" objects from the more-or-less static background. This is often done in order to identify or track the moving objects seen in the video, but the output of the background subtraction algorithm can conceivably be used for any purpose.

In its simplest form, background subtraction evaluates an inequality test at each pixel n of each frame k of a video

$$\left|I^{(k)}[n] - B[n]\right| \underset{\mathcal{B}}{\overset{\mathcal{F}}{\gtrless}} \theta \tag{3.1}$$

where $I^{(k)}[n]$ is the value of n seen in frame k (intensity on a scale from 0 to 255 for greyscale video, colour for full-colour video, etc.), $B[n]$ represents the known or estimated value of the true background at n and θ is some chosen threshold [22]. Evaluating this inequality will assign n either to the foreground set $\mathcal{F}$ or to the background set $\mathcal{B}$.

If B and I are known perfectly, this problem becomes trivial, but the fact that Equation (3.1) is not simply a test on whether $I^{(k)}[n] = B[n]$ hints at the true difficulties involved. In practice, background subtraction must account for several factors including: a background which is not completely static (e.g., due to the slight fluttering of leaves on trees), noise in the image (e.g., due to camera jitter) and changes to the background over time (e.g., as the sun slowly sets). Using a non-zero threshold in Equation (3.1) can help compensate for some of these factors, but determining the correct model for B and the correct value for θ can still be quite difficult. Fortunately, a myriad of background subtraction algorithms have already been developed to cope with these problems. In the following sections, we describe three such methods.

3.2 Background Subtraction with Temporal Background Modelling

The most basic method we consider in this chapter highly resembles Equation (3.1). This method seeks only to model the distribution of the background, specifically by using a nonparametric Gaussian kernel density function [23] and by relying on the concept of temporal similarity, the assumption that if a pixel is in the background, measurements taken on it within a short time frame should be similar to one another. Note that this temporal similarity is not expected to hold over long periods of time, which allows the background to evolve, as long as it does so more slowly than the foreground. More formally, this model dictates that the background PDF $P_{\mathcal{B}}$ of the intensity $I^{(k)}[n]$ of a pixel n of a frame k of video can be modelled as

$$P_{\mathcal{B}}\left(I^{(k)}[n]\right) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{K}\left(I^{(k)}[n] - I^{(k-i)}[n]\right) \tag{3.2}$$

where $\mathcal{K}(\cdot)$ is a zero-mean Gaussian kernel function with a variance σ_t^2 which is assumed to be constant. Here, N refers to the number of recent frames used to model the background at frame k.

Once $P_{\mathcal{B}}$ has been formed in this way, a simple test

$$\frac{P_{\mathcal{B}}\left(I^{(k)}[n]\right)}{P_{\mathcal{F}}\left(I^{(k)}[n]\right)} \underset{\mathcal{B}}{\overset{\mathcal{F}}{\lesseqgtr}} \eta\frac{\pi_{\mathcal{F}}}{\pi_{\mathcal{B}}} \tag{3.3}$$

is carried out to determine if each pixel in each frame is in the background. In this equation, η is a cost term which can be used to penalize or favour different classification errors (false positives, etc.) and $P_{\mathcal{F}}$ is the foreground PDF which is assumed to be constant for the sake of simplicity; this assumption will be modified by the more complicated background subtraction algorithms discussed in Sections 3.3 and 3.4. Likewise, for now, the prior probabilities that a pixel is in the background or the foreground, $\pi_{\mathcal{F}}$ and $\pi_{\mathcal{B}}$, are also considered to be constant and equal to one another such that $\pi_{\mathcal{F}}/\pi_{\mathcal{B}} = 1$. The four constants in Equation (3.3) (η, $P_{\mathcal{F}}$, $\pi_{\mathcal{F}}$ and $\pi_{\mathcal{B}}$) can be combined into a single new constant θ, turning the equation into

$$P_{\mathcal{B}}\left(I^{(k)}[n]\right) \underset{\mathcal{B}}{\overset{\mathcal{F}}{\lesseqgtr}} \theta. \tag{3.4}$$

The beauty of background subtraction with temporal background modelling (TBM) lies in its simplicity. Compared to the other background subtraction algorithms which we will present, its computational complexity is low and it has relatively few parameters which need to be tuned (N, σ_t^2 and θ).

3.3 Foreground-Adaptive Background Subtraction

Foreground-adaptive background subtraction (FABS) [22, 24] employs the same methods described in Section 3.2 to model $P_{\mathcal{B}}$. It then goes on to use spatial similarity—the idea that foreground pixels in the same image which are close together will tend to be similar in colour or intensity—to model $P_{\mathcal{F}}$.

More specifically, McHugh et al. [24] employ the concept of a neighbourhood $\mathcal{N}(n)$: a square of a certain size centered on n, inside of which all foreground pixels will tend to be similar. It is easy to see why this would be so. For instance, say we know that pixel n is part of the face of a person walking in the foreground. We can guess that any pixels in the neighbourhood of n which are also in the foreground will most likely be flesh-coloured as well. On the other hand, if a pixel in n's neighbourhood is a radically different colour, it lends weight to the hypothesis that it is not part of the foreground: pixels which are close to n but which are, say, green, may be part of foliage in the background.

To go about modelling the foreground, the TBM test in Equation (3.4) is first applied to a given frame to assign a preliminary background or foreground label to each pixel. Then, a first model of the foreground, $P_{\mathcal{F}_0}$, is built according to

$$P_{\mathcal{F}_0}\left(I^{(k)}[n]\right) = \frac{1}{\left|\mathcal{N}^{(k)}_{\mathcal{F}_0}(n)\right|} \sum_{m \in \mathcal{N}^{(k)}_{\mathcal{F}_0}(n)} \mathcal{K}\left(I^{(k)}[n] - I^{(k)}[m]\right) \tag{3.5}$$

where $\mathcal{N}_{\mathcal{F}_0}(n)$ is a set containing all the neighbours of n which have been assigned a foreground label by the preliminary test. Once again, $\mathcal{K}(\cdot)$ is a zero-mean Gaussian kernel function, this time with a variance σ_s^2, which is also assumed to be constant. Once $P_{\mathcal{F}_0}$ has been calculated, it is used in

$$\frac{P_{\mathcal{B}}\left(I^{(k)}[n]\right)}{P_{\mathcal{F}_0}\left(I^{(k)}[n]\right)} \underset{\mathcal{B}}{\overset{\mathcal{F}}{\lessgtr}} \eta \tag{3.6}$$

to assign new labels to each pixel. These labels can then be used to build $\mathcal{N}_{\mathcal{F}_1}$, $P_{\mathcal{F}_1}$, $\mathcal{N}_{\mathcal{F}_2}$, $P_{\mathcal{F}_2}$, ... as desired. McHugh [22] suggests that only 2–3 iterations of this process are usually needed for the labels to stabilize to what can then be called $P_{\mathcal{F}}$ (with no iterative subscript).

While the sets of background pixels and foreground pixels returned by FABS should be closer to the ground truth than those returned by TBM, it should be noted that FABS introduces an element of positive feedback to the background subtraction process. FABS is specifically designed to look for foreground pixels which may have erroneously been left in the background by a simpler background subtraction algorithm. These pixels are then added to $\mathcal{F}$. This means that if the value of η is set too high, FABS may eventually place every pixel in the image in $\mathcal{F}$. The threshold η must therefore be chosen quite carefully in order to avoid this scenario.

We point out here that FABS is more computationally-complex than TBM. It also introduces several more parameters which must be set, namely η, σ_s^2 and neighbourhood width/length.

3.4 Foreground-Adaptive Background Subtraction with Markov Modelling of Change Labels

Foreground-adaptive background subtraction with Markov modelling of change labels (FABS-MMCL) [22, 24] employs the aforementioned methods in order to model $P_{\mathcal{B}}$ and $P_{\mathcal{F}}$ and then uses spatial ergodicity—the idea that pixels which are close together will tend to belong either to the foreground or the background together—to model $\pi_{\mathcal{F}}$ and $\pi_{\mathcal{B}}$, thereby completing Equation (3.3).

In order to do this, McHugh et al. [24] propose that the FABS test in Equation (3.6) first be used to label each pixel in a given frame. McHugh [22] then points out that, by assuming the pixels' true labels obey the Markov property, these initial labels can be thought of as a single realization of a Markov random field M. The a priori probabilities of such a field taking a certain value follow the Gibbs distribution

$$P(M = m) = \frac{1}{Z} \exp\left(\frac{-1}{\gamma} \sum_{q \in \mathcal{Q}} V(q)\right) \tag{3.7}$$

where Z is a normalization constant (which will eventually be cancelled out when $\pi_{\mathcal{F}}$ is divided by $\pi_{\mathcal{B}}$) and γ is the natural temperature of the distribution. $V(\cdot)$ is a potential function which operates on $\mathcal{Q}$, the set of cliques in the image, which McHugh defines to be the set of 2-element cliques of the second-order Markov neighbourhood. This means that

$$\sum_{q \in \mathcal{Q}} V(q) = \sum_{\{n,n'\} \in \mathcal{Q}} V(n, n'), \tag{3.8}$$

where $\mathcal{Q}$ corresponds to a pixel and its eight immediate neighbours.

Relying on the aforementioned concept of spatial ergodicity, McHugh further sets V to be the Ising potential function

$$V(n, n') = \begin{cases} 0 & \text{if pixels } n \text{ and } n' \text{ have the same label} \\ 1 & \text{otherwise} \end{cases}, \tag{3.9}$$

since this penalizes sections of a frame which contain many different labels i.e., which display a high degree of non-ergodicity. This discourages the presence of isolated "false positives" (which, in this case, correspond to false assignments of the foreground label).

By temporarily assuming that M is known for all pixels except for n, these formulations can be combined with Equation (3.3) to create

$$\frac{P_{\mathcal{B}}\left(I^{(k)}[n]\right)}{P_{\mathcal{F}}\left(I^{(k)}[n]\right)} \underset{\mathcal{B}}{\overset{\mathcal{F}}{\lessgtr}} \eta \exp\left(\frac{1}{\gamma}\left[\sum_{\{n,n'\} \in \mathcal{Q}} V(n, n')\right]\Bigg|_{n \in \mathcal{B}} - \frac{1}{\gamma}\left[\sum_{\{n,n'\} \in \mathcal{Q}} V(n, n')\right]\Bigg|_{n \in \mathcal{F}}\right), \tag{3.10}$$

which—letting our previously-described neighbourhoods be of size 3×3—can be simplified to

$$\frac{P_{\mathcal{B}}\left(I^{(k)}[n]\right)}{P_{\mathcal{F}}\left(I^{(k)}[n]\right)} \underset{\mathcal{B}}{\overset{\mathcal{F}}{\lessgtr}} \eta \exp\left(\frac{1}{\gamma}\left(\left|\mathcal{N}_{\mathcal{F}}^{(k)}(n)\right| - \left|\mathcal{N}_{\mathcal{B}}^{(k)}(n)\right|\right)\right) \tag{3.11}$$

where $\left|\mathcal{N}_{\mathcal{F}}^{(k)}(n)\right|$ and $\left|\mathcal{N}_{\mathcal{B}}^{(k)}(n)\right|$ are the number of foreground and background neighbours of n in frame k as labelled by the first application of the FABS test in Equation (3.6) and then as modified by the FABS-MMCL test in Equation (3.11) itself. Note that this assumption (that M is known for all pixels except n) is a natural one if Equation (3.11) is applied iteratively, and, indeed, McHugh recommends that the technique known as iterated conditional modes [25] be used to optimize the final labels. This requires that the FABS-MMCL test be repeated several times over all pixels in turn: as each pixel is evaluated, its label is changed in-place (if necessary) before the next pixel is evaluated. The direction in which this evaluation proceeds alternates to avoid biasing the final labels. Although the recommended number of times each pixel must be evaluated is relatively low (≈ 10), FABS-MMCL is clearly the most computationally-intensive of the three methods. It also has the most parameters which need to be set.

FABS-MMCL is designed in part to prune back the effects of the overly-greedy FABS algorithm by reducing erroneous foreground labelling. Because of this, the values of parameters η, σ_s^2 and neighbourhood size which produce the best results for FABS in isolation may not be optimal when the outputs of this algorithm are intended to be fed into FABS-MMCL. Used alone, if FABS is permitted to overestimate the foreground set $\mathcal{F}$, this will decrease the performance of this algorithm. However, if this $\mathcal{F}$ is then used to initialize FABS-MMCL (which is capable of pruning away the erroneous foreground pixels), we may end up with a final result that is better than the one we would have obtained had FABS-MMCL been initialized with a smaller $\mathcal{F}$ which was missing one or more foreground pixels.

The three background subtraction algorithms we have discussed in this chapter have already been used successfully in the computer vision and image processing domains. In the next chapter, we adapt them for online calibration of RF sensing networks.

Chapter 4

Adaptation of Background Subtraction to WSNs

Having now introduced three different background subtraction methods—TBM, FABS and FABS-MMCL—we now wish to adapt them to the domain of WSNs. In Section 4.1, we explain why background subtraction is particularly well-suited to estimating baseline RSS in the first place. Then, in Section 4.2 we explain how we adapt TBM to the domain of WSNs, including how to reinterpret temporal similarity and image-processing- and computer-vision-specific concepts such as frames, pixels and intensities. In Section 4.3, we discuss how to reinterpret the concept of spatial similarity which underlies FABS. Finally, in Section 4.4, we discuss how to reinterpret the concept of spatial ergodicity which underlies FABS-MMCL.

4.1 The Difficulties of Determining Baseline RSS

We have already seen the importance of estimating the baseline RSS in Chapter 2. We have also seen that existing RF tomography algorithms do not consider how this value can be obtained in the absence of a calibration period. It is worth pausing, then, to ask whether any simple techniques can be used to determine this value when it cannot be measured directly.

One possible naive approach to determining the baseline RSS for a given link is simply to average over all available RSS measurements for that link. However, if an obstruction is moving around or across the link, this average will incorporate the changes it causes. In this case, we can only hope that enough measurements will be taken during times when the effect of the obstruction is negligible, so that this effect will not dominate the final average. Although this simple mean approximation (MA) approach is straightforward and easy to implement, we certainly cannot guarantee that the values produced by this algorithm will converge to the baseline RSS if the obstruction does not behave randomly. For instance, if the obstruction spends most of the observation period travelling along the direct LOS between two nodes, the vast majority of the measurements taken on that link will be perturbed from the baseline RSS. Furthermore, even if the obstruction does move around more freely, if the environment it is moving in experiences severe multipath effects, the majority of the measurements taken may be perturbed anyway.

Another naive approach might be to take the median of the available RSS measurements on each link, as this statistic is often more robust to perturbations than the mean [18]. However, this approach is only viable under the assumption that each link will be in the background for at least half of the observation period [26]. Again, an improperly-behaving obstruction or severe multipath effects might cause this assumption to be violated.

Yet another simplistic approach might be to examine the RSS for a given link over a window of time, searching for periods when the RSS drops suddenly. One might then attribute these drops to the attenuation caused by a newly-appearing obstruction, use a threshold to discard these lowered values as outliers and then take the mean of the remaining measurements. As we discussed in Section 2.2, though, the effect of an obstruction on RSS is not that straightforward; recall that obstructions can just as easily cause the RSS on a link to increase. In addition, noise, decreasing battery voltage and shifts in the nodes' positions will also cause changes to the RSS, but we will probably not wish to discard measurements which are perturbed for these reasons. These factors drastically reduce the performance of simple techniques such as thresholding [27].

4.1.1 The Benefits of Background Subtraction

These naive algorithms are not equipped to deal with the complex behaviour of wireless signals. However, we can see strong parallels between the problem of determining which links in a network are currently being affected by a moving obstruction (i.e., are in the foreground) and the problem of determining which pixels in an image are currently displaying a moving foreground object. Indeed, the aforementioned naive thresholding method can perhaps be seen as a type of highly-simplified background subtraction where the background is modelled as the RSS values on the upper end of the range of measurements seen on each link. Fortunately, as discussed in Chapter 3, better background subtraction algorithms—which are specifically designed to deal with noisy measurements and with backgrounds which fluctuate both over the long and short term—already exist. By forming an analogy between the intensity of a pixel in a frame of greyscale video and the RSS measured on a link at a given time, we can hypothesize that background subtraction should be able to identify which links in a network are in the foreground or background at any given time. We would then be able to estimate the baseline RSS on each link more accurately by discarding any measurements taken on that link at times when the it is in the foreground.

For our purposes, the TBM algorithm is preferred due to its low complexity. Furthermore, because it relies completely on the temporal characteristics of the relevant measurements while ignoring their spatial properties, it can be used for node localization and in other applications where the nodes' locations are unknown.

However, as previously mentioned, we are aware of the large variance in RSS measurements seen even on background links. This leads us to hypothesize that the assumption of temporal similarity which underlies TBM may not always hold as strongly as we might hope. To account for this possibility, we also test the FABS and FABS-MMCL methods in the hopes that their use of spatial characteristics will further improve our estimates of the baseline RSS. While the extra knowledge required to use FABS and FABS-MMCL techniques means that they are unsuited to node localization applications, they can still be used for RF tomography applications where the locations of the nodes are known.

4.2 Basic Background Subtraction in WSNs

Before any method of background subtraction can be applied to a WSN, it is necessary to establish what constitutes a "pixel," a measurement of "intensity" and a "frame" in this context. We can immediately make the easy association between a single pixel n in an image and a single bidirectional link ℓ in a WSN. Furthermore, we can see that the discretized value of the RSS (in dBm) measured on a link during an as-yet-to-be-completely-defined frame k, $R^{(k)}[\ell]$, represents a natural analogue to the greyscale intensity of a pixel, $I^{(k)}[n]$.

In the computer vision and image processing domains, a single frame of video comprises a single measurement for each pixel in the video image. Ideally, we would like our corresponding RSS frame to contain a single RSS measurement for each bidirectional link as well. Also ideally, all these measurements would be taken at exactly the same time. As discussed in Section 2.1, this is simply not possible, but we can build upon the aforementioned strategy of collecting several consecutive measurement vectors and treating them as though they were taken at the same time to form the basis for our RSS frames. This leaves only the precise number of vectors which must be combined together in this way left to be determined.

At a minimum, in order to ensure that we have a measurement for every link in our network, we must wait for each of our Φ nodes to transmit once. But, because WSNs are subject to dropped packets, a single measurement cycle of this kind may not actually return a measurement for each link. If this happens, we might need to wait for one or more additional measurement cycles in order to complete our frame, with the time needed to do so doubling, tripling, quadrupling...and with the chances of the network conditions remaining the same for the length of the frame constantly decreasing. Hence, instead of a frame being comprised of a single measurement for each link, we define a frame as being built from Φ consecutive unidirectional measurement vectors. If this window contains more than one measurement for a given unidirectional link (which would occur if a dropped packet causes it to contain *no* measurements for some other link), these measurements are averaged together. Once this is done, the two unidirectional measurements for each bidirectional link are averaged together as well to create a single measurement for each bidirectional link. Finally, if measurements are not available for one or more links, these values are filled in using the values from the previous frame. This is not ideal, but it is better than the alternative of waiting for another Φ transmissions before the frame can be constructed. This process creates a single frame of $\Lambda = \frac{\Phi^2 - \Phi}{2}$ measurements. The next frame is then created from the next Φ consecutive unidirectional measurement vectors and so on. A consequence of this method is that there is no overlap in the measurements used to form each frame, except where dropped packets force us to borrow measurements from a previous frame.

Having established these analogies for pixels, pixel intensities and frames, we see that the temporal similarity underlying TBM should still hold here to some degree: if the effects of noise are small, the consecutive measurements taken on a given link should not vary, as long as that link stays in the background (discounting long-term changes such as those caused by a dying battery). In that case, we can calculate the background PDF $P_{\mathcal{B}}$ via an updated version of Equation (3.2) such that

$$P_{\mathcal{B}}\left(R^{(k)}[\ell]\right) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{K}\left(R^{(k)}[\ell] - R^{(k-i)}[\ell]\right) \tag{4.1}$$

and we can build our sets of background and foreground links $\mathcal{B}$ and $\mathcal{F}$ via an updated version of Equation (3.4) such that

$$P_{\mathcal{B}}\left(R^{(k)}[\ell]\right) \underset{\mathcal{B}}{\overset{\mathcal{F}}{\lessgtr}} \theta. \tag{4.2}$$

Recall that N is the number of recent frames used to model the background at frame k and $R^{(k)}[\ell]$ is the measured RSS value for link ℓ at frame k.

On the other hand, if the effects of noise cannot easily be ignored, we may be able to improve the results of background subtraction by using the FABS algorithm, which—in the computer vision and image processing domains, at least—makes use of the concept of spatial similarity to refine the results of TBM.

4.3 Spatial Similarity in WSNs

While determining analogues for pixels, pixel intensity and frames is relatively straightforward, applying spatial similarity and the concept of neighbourhoods to WSNs is more complicated. This is largely because there is no reason why two foreground links which are close together must have similar RSS values. For instance, imagine a network in which nodes are laid out around the perimeter of a square. Let ℓ_1 be the link which connects nodes located at $(0,0)$ and $(0,1)$, and let ℓ_2 be the link which connects nodes located at $(0,0)$ and $(0,7)$. Furthermore, let both these links be obstructed by the same object. The two overlapping links are indeed physically close together, but because ℓ_1 is 1 m long while ℓ_2 is 7 m long, the log-distance path loss model (Equation (2.3)) suggests that their RSS values will be quite different from one another. From the knife-edge diffraction model (Equation (2.5)), we can also see that the obstruction will affect both links differently as well. Therefore, if we wish for all foreground links in a given neighbourhood to have similar RSS values, we cannot simply define the neighbourhood of ℓ to be links which are close to ℓ.

Instead, we create a type of neighbourhood, $\mathcal{L}(\ell)$, such that $\mathcal{L}(\ell)$ contains links which are similar in length to ℓ. Recalling that d_ℓ represents the length of link ℓ, $\mathcal{L}(\ell)$ is thus defined as

$$\mathcal{L}(\ell) = \{\ell' : |d_\ell - d_{\ell'}| \leq \tau\} \tag{4.3}$$

where $\tau \geq 0$ is left as a parameter to be tuned. Note that this definition also implies the creation of $\mathcal{L}_{\mathcal{F}}^{(k)}(\ell)$, which represents the set of foreground neighbours associated with $\mathcal{L}(\ell)$ in frame k.

When formulating $\mathcal{L}(\ell)$ in this way, we are aware that not all links in this type of neighbourhood are guaranteed to have similar RSS values. Even for links of the exact same length, if a 1 m link in the network passes through a wall while another passes through free space, their values will be different. Keep in mind, though, that even in the computer vision and image processing domains, not all foreground pixels in a neighbourhood are strictly guaranteed to have a similar intensity (e.g., a flesh-coloured pixel may be right next to a blue pixel which depicts a shirt).

Furthermore, from the knife-edge diffraction model, we are aware that two links of the same length whose LOS paths are both obstructed by the same moving object may have different RSS values if the object intersects them in different ways, e.g., if an obstruction is located at the extreme end of one link while simultaneously being in the center of the other [19]. In practice, though, we find that this difference is within the range of the normal variations due to noise seen on the links. We therefore feel comfortable using the mechanism we have just described in order to construct $\mathcal{L}(\ell)$.

In this way, spatial similarity in the image-processing and computer-vision domains becomes similarity-of-length in the WSN domain.

We can now present our new expressions for the model of the preliminary foreground PDF $P_{\mathcal{F}_0}$ (originally seen in Equation (3.5)) and for the FABS test (originally seen in Equation (3.6)) such that

$$P_{\mathcal{F}_0}\left(R^{(k)}[\ell]\right) = \frac{1}{\left|\mathcal{L}^{(k)}_{\mathcal{F}_0}(\ell)\right|} \sum_{m \in \mathcal{L}^{(k)}_{\mathcal{F}_0}(\ell)} \mathcal{K}\left(R^{(k)}[\ell] - R^{(k)}[m]\right) \tag{4.4}$$

and

$$\frac{P_{\mathcal{B}}\left(R^{(k)}[\ell]\right)}{P_{\mathcal{F}}\left(R^{(k)}[\ell]\right)} \underset{\mathcal{B}}{\overset{\mathcal{F}}{\lessgtr}} \eta. \tag{4.5}$$

4.4 Spatial Ergodicity in WSNs

We also consider the concept of spatial ergodicity which underlies the FABS-MMCL method. Fortunately, spatial ergodicity carries over to the WSN domain more smoothly—though not completely unchanged—than does spatial similarity. We can still say that if two links are close enough together—i.e., they are neighbours—they are both likely to pass through or avoid the same obstructions and hence be in the foreground or background together. Nonetheless, we will have to specify what exactly it means for two links to be "close enough together." Do they have to share a common endpoint? Do their LOS paths have to overlap in space? Intersect? Pass within a certain distance of one another?

4.4.1 Rectangle-Based Neighbourhoods

We present two different methods for determining whether two links are close enough together for spatial ergodicity to hold. The first of these methods involves partitioning the interior of the region of interest into rectangles whose side lengths are determined by the node separations. These rectangles are similar to the cubic voxels proposed by Wilson and Patwari [7]. We denote by $\rho(\ell)$ the set of rectangles through which a link ℓ passes. Then, for two links, ℓ_1 and ℓ_2 with associated sets of rectangles $\rho(\ell_1)$ and $\rho(\ell_2)$, we define

$\chi(\ell_1, \ell_2)$, the percentage of rectangles common to both links as

$$\chi(\ell_1, \ell_2) = \frac{|\rho(\ell_1) \cap \rho(\ell_2)|}{\min(|\rho(\ell_1)|, |\rho(\ell_2)|)} \times 100. \tag{4.6}$$

Finally, if this percentage is larger than a certain tuneable threshold, we deem ℓ_1 and ℓ_2 to be neighbours. A graphical representation of this process can be seen in Figure 4.1. Note that a percentage formulation like this one must be used (as opposed to dividing $|\rho(\ell_1) \cap \rho(\ell_2)|$ by $\max(|\rho(\ell_1)|, |\rho(\ell_2)|)$ or by 1) in order to prevent short links from having artificially-few neighbours. We refer to this method as FABS-MMCL-R.

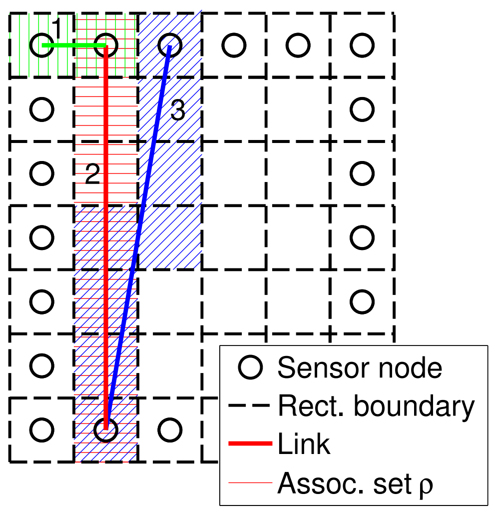


Fig. 4.1 A graphical representation of the process used to build FABS-MMCL-R neighbourhoods. The set of rectangles ρ associated with each link is clearly marked. We can see that $|\rho(\ell_1)| = 2$, that $|\rho(\ell_2)| = 7$ and that $|\rho(\ell_1) \cap \rho(\ell_2)| = 1$. This represents a shared rectangle percentage of $\chi(\ell_1, \ell_2) = \frac{1}{\min(2,7)} \times 100 = 50\%$. Similarly, the shared rectangle percentage between ℓ_2 and ℓ_3 is $\chi(\ell_2, \ell_3) = \frac{4}{\min(7,8)} \times 100 = 57\%$.

Unlike in the computer vision and image processing domains, the WSN-domain neighbourhoods used for FABS and FABS-MMCL are not the same sets. As previously mentioned, we denote the neighbourhood for FABS as $\mathcal{L}(\ell)$. We now denote the neighbourhood used for FABS-MMCL as $\mathcal{S}(\ell)$. For FABS-MMCL-R, this neighbourhood is defined as

$$\mathcal{S}(\ell) = \{\ell' : \chi(\ell, \ell') \geq J\} \tag{4.7}$$

where $0 < J \leq 100$ is left as a parameter to be tuned. We also define $\mathcal{S}_{\mathcal{F}}^{(k)}(\ell)$ and $\mathcal{S}_{\mathcal{B}}^{(k)}(\ell)$, the sets of foreground and background neighbours, respectively, associated with $\mathcal{S}(\ell)$ in frame k.

This leads to the FABS-MMCL test in Equation (3.11) becoming

$$\frac{P_{\mathcal{B}}\left(R^{(k)}[\ell]\right)}{P_{\mathcal{F}}\left(R^{(k)}[\ell]\right)} \underset{\mathcal{B}}{\overset{\mathcal{F}}{\lessgtr}} \eta \exp\left(\frac{1}{\gamma}\left(\left|\mathcal{S}_{\mathcal{F}}^{(k)}(\ell)\right| - \left|\mathcal{S}_{\mathcal{B}}^{(k)}(\ell)\right|\right)\right). \tag{4.8}$$

4.4.2 Obstruction-Based Neighbourhoods

Although FABS-MMCL-R is fairly intuitive, it has its problems. For instance, if two LOS links have a small angle of separation, they may indeed be close enough to cross the same rectangles over a large fraction of their length, thereby qualifying as neighbours. However, this is misleading if their paths diverge significantly by the time one of them crosses the spot where the obstruction is located, as seen in Figure 4.2. We therefore also propose another method for determining $\mathcal{S}(\ell)$, this one inspired by the fact that we want links to be in the same neighbourhood as each other only if they are close together when they are also close to any obstructions which may be present in the region of interest.

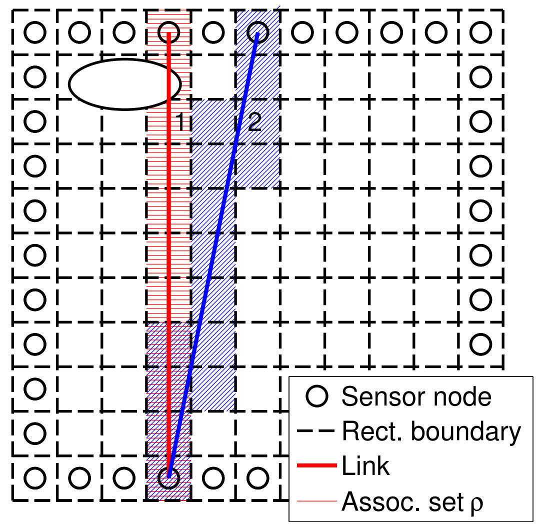


Fig. 4.2 A possible relationship between an obstruction and two different links. Links ℓ_1 and ℓ_2 may have a high enough common rectangle percentage to qualify as neighbours, but at the top of the figure, they are no longer as close together, and the pictured obstruction may not have any effect on ℓ_2.

Recall that we do not have any a priori information about the location of these obstructions. We can still hypothesize, though, that their locations may be inferred by looking for rectangles which contain particularly large numbers of foreground links. Specifically, just before each application of Equation (4.8), we can count the number of foreground links passing through each rectangle in $\rho(\ell)$. We can then find the rectangle $r \in \rho(\ell)$ with the largest number of foreground links passing through it and assign $\mathcal{S}^{(k)}(\ell)$ to be the set of links which pass through r. This is shown graphically in Figure 4.3. We refer to this method as FABS-MMCL-O.

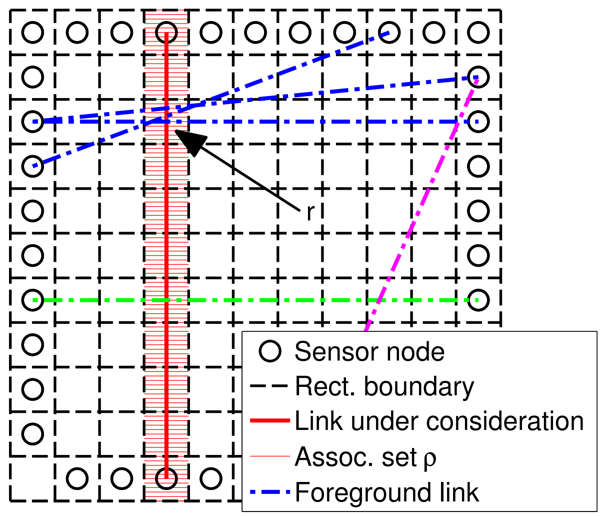


Fig. 4.3 A graphical representation of the process used to build FABS-MMCL-O neighbourhoods. Link ℓ is shown in red while the links which are currently labelled as being in the foreground are assigned various colours depending on which rectangle in $\rho(\ell)$ they pass through. We can see that one of these rectangles contains 3 foreground links (shown in blue), one contains 1 (shown in green), and the others contain 0. Since $\max(3, 1, 0) = 3$, r is the rectangle containing 3 foreground links. The neighbourhood of ℓ for the current frame, $\mathcal{S}^{(k)}(\ell)$, then consists of all the links which pass through r, and $\mathcal{S}_{\mathcal{F}}^{(k)}(\ell)$ is precisely the set of links shown in blue.

Of course, r may not be located near an obstruction at all since, in fact, it is quite possible that none of the rectangles in $\rho(\ell)$ are near the obstruction (e.g., if the LOS of ℓ is positioned entirely on the extreme east side of the network while the obstruction is on the west side of the network). If this is the case, even though r may contain more foreground links than any other rectangle in $\rho(\ell)$, $\left|\mathcal{S}_{\mathcal{F}}^{(k)}(\ell)\right|$ is still likely to be small. This will cause the exponent term in the FABS-MMCL test (Equation (4.8)) to be highly negative, decreasing the right hand side of the equation. In this way, no harm is done if r is artificially deemed to contain an obstruction. However, in our preliminary experiments with the TBM method, we notice that even when we know for a fact that r contains an obstruction, and even when our classification of foreground and background links results in an accurate estimate of the baseline RSS, there may still be relatively few foreground links and hence many background links passing through r. This means that, if we apply FABS-MMCL-O to the output of the

TBM algorithm as described, the exponential term in Equation (4.8) would almost always be highly-negative.

To counteract this tendency, we introduce a term μ to the FABS-MMCL test such that

$$\frac{P_{\mathcal{B}}\left(R^{(k)}[\ell]\right)}{P_{\mathcal{F}}\left(R^{(k)}[\ell]\right)} \underset{\mathcal{B}}{\overset{\mathcal{F}}{\lessgtr}} \eta \exp\left(\frac{1}{\gamma}\left(\left|\mathcal{S}_{\mathcal{F}}^{(k)}(\ell)\right| - \left|\mathcal{S}_{\mathcal{B}}^{(k)}(\ell)\right| + \mu\right)\right). \tag{4.9}$$

Recall that $\mathcal{B}$ and $\mathcal{F}$ are the sets of background and foreground links respectively. Then, by resetting $\mu = |\mathcal{B}| - |\mathcal{F}|$ before each application of Equation (4.9), we ensure that the distribution of the exponent is centered more-or-less around 0 instead of always being negative.

4.4.3 Other Methods of Building Neighbourhoods

Although we present two of our own models for building $\mathcal{S}^{(k)}(\ell)$, we are aware that researchers have previously considered related questions. For instance, in [28], Gudmundson proposes a model for quantifying the correlation of the shadowing observed on different links, but this model only applies to links which have at least one common endpoint, which is not the case with most of the links in a WSN. Wang et al. [29] extend Gudmundson's work to model the shadowing correlation which exists between two links, ℓ_1 and ℓ_2, when ℓ_2 is created by moving the nodes which form ℓ_1 to a new location. We might theoretically apply this model to a case where ℓ_1 and ℓ_2 are two links defined by four completely separate senders and receivers, but Wang et al.'s model also assumes that the distance between ℓ_1 and ℓ_2 will be small compared with the lengths of the links themselves. Again, this does not apply to most of the links in a WSN. Finally, Agrawal and Patwari [30,31] propose the network shadowing (NeSh) model to calculate the shadowing correlation seen across links in a multi-hop wireless network. Namely, the covariance seen between the shadowing X_1 and X_2 on two links ℓ_1 and ℓ_2 can be modelled as

$$\operatorname{Cov}(X_1, X_2) = \frac{\sigma_X^2}{\delta d_{\ell_1}^{1/2} d_{\ell_2}^{1/2}} \int_{Y_1}\int_{Y_2} e^{-\frac{||\beta-\alpha||}{\delta}} \mathrm{d}\alpha^T \mathrm{d}\beta \tag{4.10}$$

where δ is a space constant, σ_X^2 is the variance of the shadow fading, Y_1 and Y_2 are the LOS paths of ℓ_1 and ℓ_2 respectively and d_{ℓ_1} and d_{ℓ_2} are the lengths of those paths. While this method yields predictions which agree nicely with experimental results, it is relatively complex since it requires that both δ and σ_X^2 (which is likely to change over time in our case) be estimated before it can be applied. Furthermore, in order to use the covariance between two links to determine whether they are neighbours, we would need to introduce a third parameter to act as a threshold on this covariance. Our background subtraction algorithms already have a multitude of parameters to tune and the prospect of adding three more was daunting, prompting us to discard this method.

Inspired by the study of correlation coefficients observed for various link geometries presented in [30, 31], we also briefly mention another possible method for determining whether two links are neighbours which involves inspecting many features of the geometry of their LOS paths: their points of closest and farthest approach; their angle of separation; the length of their overlap, if any; their number of shared endpoints and whether or not they intersect. Each of these features can be compared to some preset threshold to determine if the links are neighbours. However, this method requires an even more intractable number of parameters. Therefore, we discard it as well in favour of FABS-MMCL-R and FABS-MMCL-O.

We now have all the tools we need in order to estimate the baseline RSS: we are aware of the effects that obstructions have on RSS values, we have described the theory underlying background subtraction, and we have explained how this theory can be adapted to apply to WSNs for use in obtaining estimates of the background, baseline RSS. We now test this theory with RSS measurements collected in real-world experiments in order to evaluate the performance of background subtraction in the WSN domain.

Chapter 5

Experimental Analysis of Background Subtraction Methods

In this chapter, we present the results of applying the aforementioned background subtraction methods to real data collected with a WSN testbed. In Section 5.1, we describe this testbed and the experiments which were performed with it in more detail. In Section 5.2 we discuss how we evaluate the performance of our algorithms. In Section 5.3, we present our results, and in Section 5.4, we examine how sensitive these results are to changes in the algorithm parameters. Then, in Section 5.5, we look at the relative complexity of each algorithm.

5.1 Experiment Setup

Through an association with the Beijing University of Posts and Telecommunications (BUPT) in Beijing, China, we have access to data from a number of RF sensing experiment sets carried out by BUPT students over the past year. Each set of experiments made use of the same methodology, the same hardware and the same software. Only the number of nodes and the environment in which they were deployed varied between experiment sets.

The specific hardware employed in these experiments consisted of a fleet of Texas Instruments CC2530 System-on-Chips, each of which was equipped with a built-in RSS indicator capable of measuring RSS to a resolution of 1 dBm. These motes conform to the IEEE 802.15.4 standard, transmitting in the 2.4 GHz band with a configurable output power of up to 4.5 dBm [32]. For these experiments, the nodes were set to transmit at 0 dBm. In addition to their built-in antennae, each mote was also augmented with an external antenna.

The motes were programmed to follow the token-ring protocol described in Section 2.1, and all Φ motes were programmed to communicate with one another, thereby forming a fully-connected topology which could directly measure the RSS between every pair of nodes in the network. The system was configured to probe this RSS (i.e., by having the next node in the ring send a new wireless message) approximately every 5 ms. Finally, the motes were also programmed to transmit their measured RSS values to a $(\Phi + 1)$st mote, which forwarded this data to a laptop computer via USB cable. The computer then recorded the reported values. This extra mote acted only as an intermediary to the laptop, and any RSS values on links which involve this mote are irrelevant to the experiments.

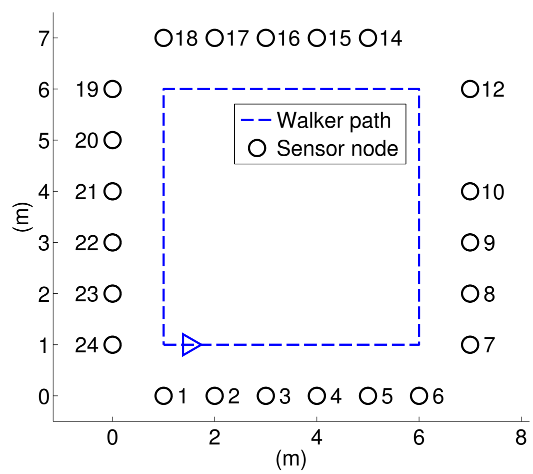
These experiments were conducted as follows: In each new environment, the nodes were mounted on ≈ 1 m-high stands around the perimeter of the region of interest. They were then allowed to transmit amongst themselves for several minutes while the network was kept vacant, gathering training data from which the baseline RSS on each link could be determined directly. After this offline calibration period was completed, a person entered the region of interest and proceeded to walk around the nodes. This person followed a path which was marked on the ground, and a timestamp was recorded as they passed various points on their route so that their location was known at all times. Each experiment set thus consists of baseline RSS data collected while the region of interest was empty, coupled with data collected during one or more "walks." Note that the length of each walk varied;

some lasted only long enough for ≈ 150 frames of measurements to be obtained, while in other cases, up to 700 frames of measurements were collected.

The first set of experiments was conducted in an outdoor field at BUPT. Outdoor environments such as fields are often the most conducive to RF tomography applications (provided the weather is fair enough not to damage the nodes, of course). In general, the lack of nearby walls and furniture means that troublesome multipath effects are far less prominent outdoors.

The node setup for the outdoor experiment set can be seen in Figure 5.1. The irregular node positioning in Figure 5.1(b) is due to the fact that nodes 11 and 13, which ought to be located in the upper right corner of the figure, were discovered to be defective after the experiments had already been completed. All the measurements taken by these nodes were therefore discarded.

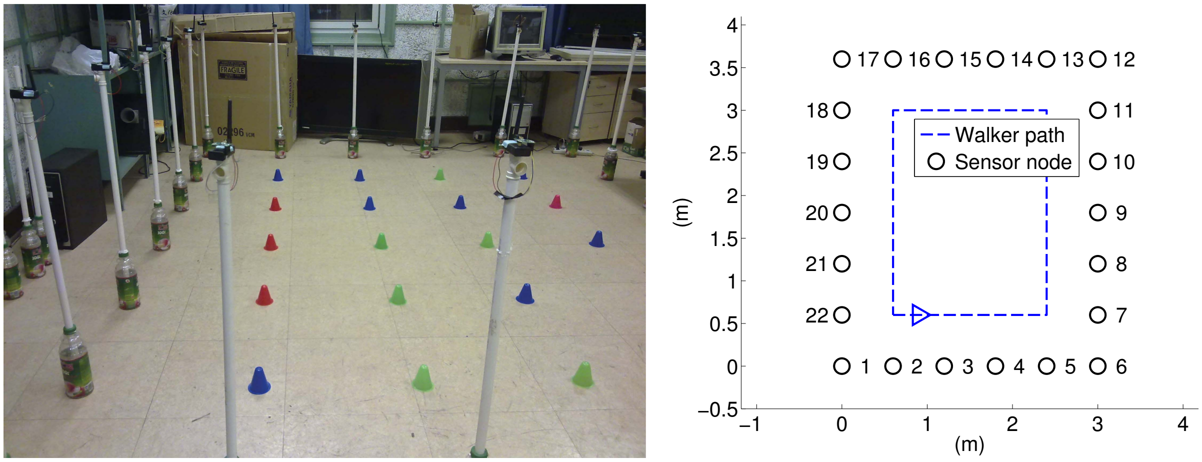
(a) Photo of the experiment setup.



(b) Diagram of the node positions.

Fig. 5.1 Node deployment for the first set of experiments, which took place in an outdoor field.

The second set of experiments was conducted in an indoor lab, with the nodes positioned around the sides of the room as seen in Figure 5.2. Although there was furniture along the walls of the room (outside the region being monitored by the nodes), no furniture was located inside the region of interest. This means that the only thing directly obstructing the LOS paths of the wireless signals at any given time was the walker sent into the region of interest; nonetheless—as in indoor environments in general—the walls, floor, ceiling and furniture still contribute to the multipath effects which generally make it harder for RF tomography applications to function properly indoors.

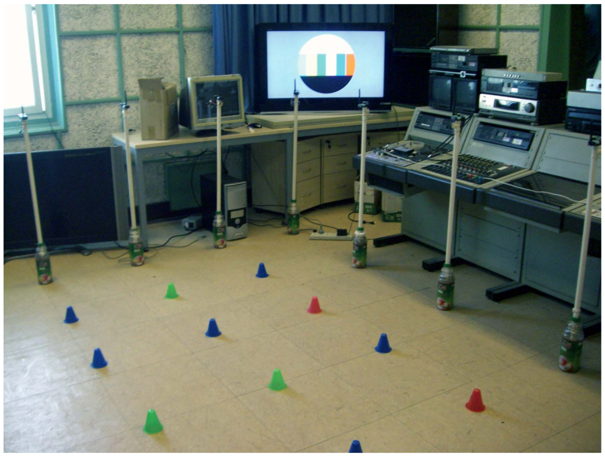


(a) Photo of the experiment setup.

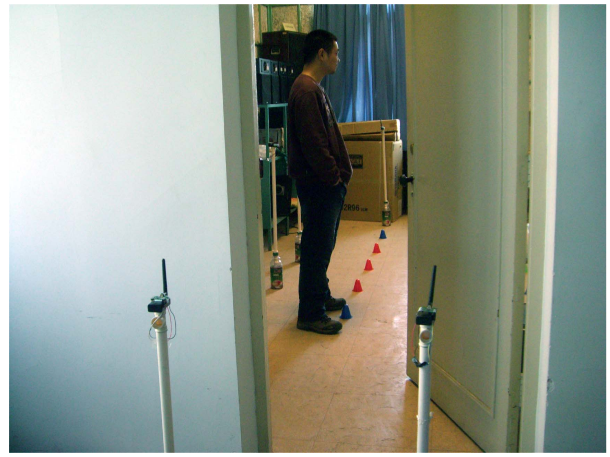
(b) Diagram of the node positions.

Fig. 5.2 Node deployment for the second set of experiments, which took place in an indoor lab.

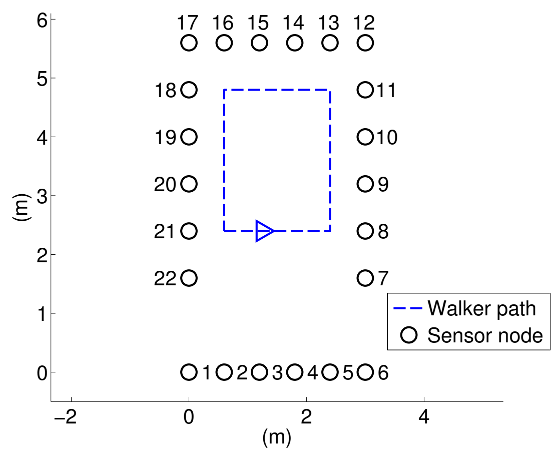
The third set of experiments was also conducted in an indoor lab. This time, several of the nodes were placed in a hallway outside the lab, as seen in Figure 5.3. They therefore monitored the inside of the lab through a solid wall which passed through the gap left between nodes 6 and 7 and nodes 1 and 22 in Figure 5.3(c). This scenario is one of the hardest for RF tomography applications to cope with. First, multipath effects in this type of environment will be significant. Furthermore, when a walker enters the region of interest, many links in the network will then be affected by multiple obstructions (the wall and the walker). As previously mentioned in Section 2.2, predicting the combined effects of multiple obstructions is incredibly complicated.



(a) Photo of the experiment setup (seen from inside the lab).



(b) Photo of the experiment setup (seen from the hall). This door was closed while the experiment was in progress.



(c) Diagram of the node positions. A wall was located in the gap between nodes 1 and 22 and nodes 6 and 7.

Fig. 5.3 Node deployment for the third set of experiments, which took place indoors, with nodes monitoring a lab through a solid wall.

The results from these experiments allow us to test the effectiveness of our background subtraction algorithms in several different environments.

5.2 Performance Evaluation Methods

The best way to evaluate the ability of our algorithm to detect foreground links would be to compare the links it assigns to the foreground set $\mathcal{F}$ with some ground truth knowledge of which links are actually in the foreground. However, there is no way to obtain this information. In general, we might assume that links whose LOS paths intersect the obstruction ought to be in the foreground, but there is no guarantee that that is the case, particularly in an indoor environment where multipath effects are non-trivial and where the NLOS components of a signal can be substantial contributors to its final RSS value. If one of these NLOS components is obstructed, the associated link may likewise be in the foreground.

We therefore evaluate our algorithms' performance in two ways: by looking directly at their ability to estimate the measured baseline RSS (as collected during the offline calibration period) and by examining the performance of RF tomographic tracking initialized with the baseline RSS values produced by the algorithms.

5.2.1 Metric 1: Comparison with the Measured Baseline RSS

Because we have access to direct measurements of the baseline RSS from the calibration period of each experiment set, we define $\tilde{R}_B[\ell]$ as the mean RSS measured on ℓ over this time period, when the region of interest was intentionally left vacant. Recall that this is the data which existing RF tomographic algorithms already assume is available to them.

Our estimate of the baseline RSS on link ℓ, $\hat{R}_B[\ell]$, is then calculated according to

$$\hat{R}_B[\ell] = \frac{\sum_{k\in K} \psi^{(k)}[\ell] R^{(k)}[\ell]}{\sum_{k\in K} \psi^{(k)}[\ell]} \tag{5.1}$$

where K is the total number of frames available at a given time and $\psi^{(k)}[\ell]$ is an indicator function defined as

$$\psi^{(k)}[\ell] = \begin{cases} 1 & \text{if } \ell \in \mathcal{B} \text{ in frame } k \\ 0 & \text{if } \ell \in \mathcal{F} \text{ in frame } k \end{cases}, \tag{5.2}$$

built using the output of our background subtraction algorithms. This $\psi^{(k)}[\ell]$ takes different values depending on whether background subtraction is carried out using TBM, FABS or

FABS-MMCL.

Once we obtain $\hat{R}_B[\ell]$, we can calculate the root mean square estimation error of the approximation (measured in dBm/link) such that

$$\text{Estimation Error} = \sqrt{\frac{1}{\Lambda}\sum_{\ell\in\Lambda}\left(\tilde{R}_B[\ell] - \hat{R}_B[\ell]\right)^2} \tag{5.3}$$

where Λ is the total number of links in the network.

This provides us with a numerical qualification for how well we have estimated the baseline RSS, which can easily be compared across our different algorithms. We can also calculate this metric for the mean approximation method described in Section 4.1, which constitutes the simplest possible way to estimate baseline RSS without access to a calibration period. Note that the MA algorithm corresponds to setting $\psi^{(k)}[\ell] = 1$ for all values of k and ℓ.

5.2.2 Metric 2: Tracking Performance

To get a more intuitive feel for the implications of a certain level of estimation error, we employ the RF tomographic tracking algorithm proposed by Li et al. and Chen [8,9]. This algorithm must be given a value for the baseline RSS on each link in the network before it can begin tracking. By comparing tracking performance when using $\tilde{\mathbf{R}}_\mathbf{B}$, the vector of measured baseline RSS values, to tracking performance when using different versions of $\hat{\mathbf{R}}_\mathbf{B}$, we can see whether our estimates of the baseline RSS are accurate enough for real-world applications.

As explained in Section 2.4, there is a certain level of randomness inherent in the tracking algorithm. Accordingly, for each value of $\tilde{\mathbf{R}}_\mathbf{B}$ or $\hat{\mathbf{R}}_\mathbf{B}$, we run $A = 100$ realizations of the algorithm and report the performance over this ensemble in the form of the root mean square tracking error which is calculated as

$$\text{Tracking Error} = \frac{1}{A}\sum_{a=1}^{A}\sqrt{\frac{1}{T}\sum_{t=1}^{T}\left|\left|\mathbf{z}_t - \hat{\mathbf{z}}_t^{(a)}\right|\right|^2} \tag{5.4}$$

where $\hat{\mathbf{z}}_t^{(a)}$ is the estimated value of the person's position at time $t \in T$ as reported by realization a.

5.2.3 Additional Evaluation

In order to further understand the behaviour of our background subtraction algorithms, we will also take a qualitative look at which links and which RSS values end up being selected for the foreground using various histograms and other diagrams.

Because this thesis constitutes the first examination of the potential of background subtraction to estimate baseline RSS in WSNs, we are mostly interested in determining the maximum improvements which can be gained from this technique. Therefore, we present the results which are obtained when the background subtraction parameters (θ, N, etc.) are specifically chosen to maximize performance. In Section 5.4, we also provide a brief examination of the algorithms' sensitivity to parameter values which differ from these optimal settings.

5.3 Results for Each Experiment Sets

5.3.1 Results for the Outdoor Experiment (Experiment Set 1)

Figure 5.4 presents the estimation error obtained when using our background subtraction algorithms to estimate baseline RSS values for an experiment conducted in an outdoor field. Table 5.1 presents the parameter values used to obtain the pictured results.

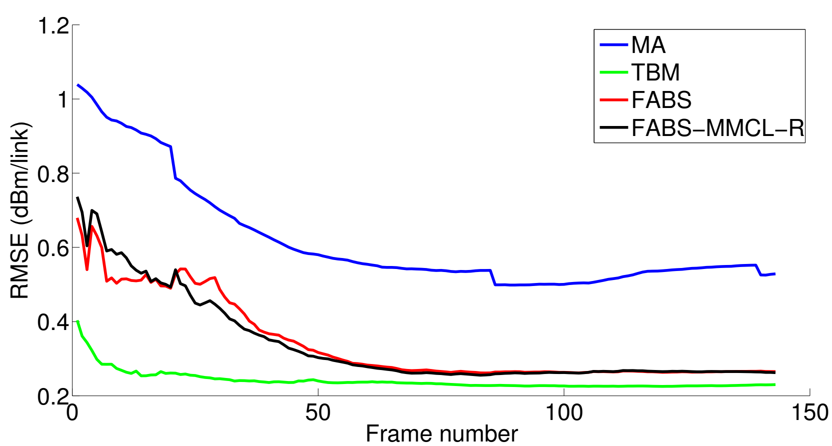


Fig. 5.4 RMSE obtained when estimating the baseline RSS for experiment set 1 (conducted in an outdoor field). The 143 frames used were collected while a walker made one round trip around the path seen in Figure 5.1(b) over a period of ≈ 20 seconds.

Parameter	TBM	FABS	FABS-MMCL-R
N	25	25	25
σ_t^2	4	4	4
θ	0.17	0.17	0.17
τ	–	0	0
σ_s^2	–	4	4
η	–	1	1
γ	–	–	5
J	–	–	95

Table 5.1 Optimal background subtraction parameter values for experiment set 1 (conducted in an outdoor field) for various algorithms.

From the estimation error results, we can see that TBM provides a sizeable improvement over MA. Recall that TBM is the least computationally-complex of all the algorithms we present and that no knowledge of the nodes' positions is required in order to apply it. Consequently, it is particularly encouraging that we can see a large improvement over MA when using this particular method. We also see that the more complicated algorithms—which we proposed in hopes of achieving even greater improvements—do not actually do any better than TBM in this case.

Since this original experiment set includes only a single "walk", we validate these results by repeating our analysis on data sets collected several months later, in a different outdoor field also located at BUPT. These results are presented in Appendix A. Note that in both fields, we see the same pattern in our results: TBM shows an improvement over MA while the more complex FABS-based approaches do not show any significant improvement compared to TBM. Additionally, it is worth noting that these similar results are obtained in two different outdoor environments with nodes set out in two slightly different patterns (since the broken nodes 11 and 13 from experiment set 1 were fixed before the experiment presented in Appendix A was carried out), but using the same parameters seen in Table 5.1. This is very encouraging, because even if these parameter values seem slightly arbitrary, this shows that at least they are transferrable across similar—though not identical—environments.

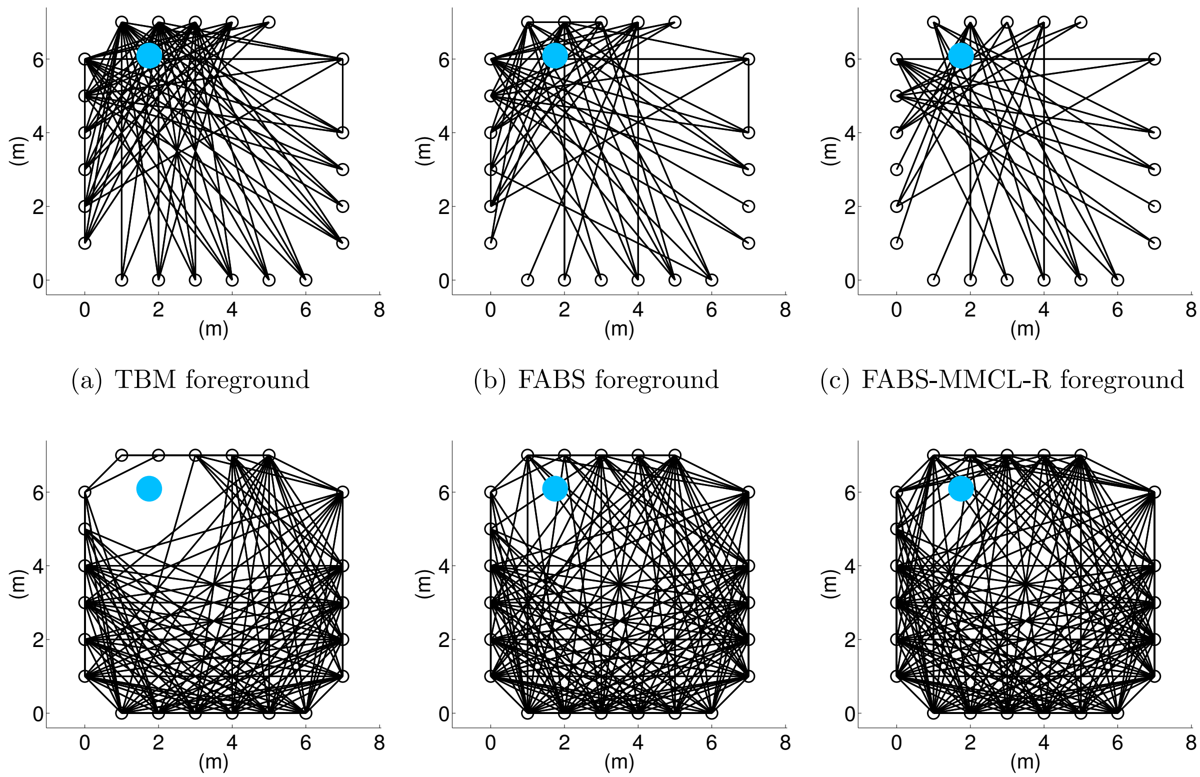
Returning to the experiment set 1 data, we now present the results of using Li et al. and Chen's tracking algorithm [8, 9] (initialized in turn with $\tilde{\mathbf{R}}_{\mathbf{B}}$ and with various values of $\hat{\mathbf{R}}_{\mathbf{B}}$) to track the movement of a walking person. These tracking errors are presented in Table 5.2.

Algorithm	Tracking Error (m)
Calibration Data	0.4073 ± 0.0655
MA	0.4363 ± 0.0711
TBM	0.4319 ± 0.0655
FABS	0.4335 ± 0.0667
FABS-MMCL-R	0.4339 ± 0.0616

Table 5.2 RMSE for tracking using experiment set 1 (conducted in an outdoor field). The tracking algorithm is initialized in turn with measured values of the baseline RSS and with various estimates of the baseline RSS.

From these results, we can see that the ideal situation (where we can initialize tracking with the directly-measured data) provides the best performance, as expected. We can also see that using MA to build $\hat{\mathbf{R}}_{\mathbf{B}}$ causes our performance to suffer slightly, while TBM provides an improvement over MA. Finally, we can also see that the more complex algorithms are no more helpful than TBM. In all cases, though, the tracking error is relatively low, as expected in an outdoor environment.

In order to understand why our more complex background subtraction approaches do not provide any improvement over TBM, we examine which links are being selected as foreground and background links by each algorithm. This can be seen (for a single frame) in Figure 5.5. Note, of course, that any link which is not in the foreground in a given frame must be in the background, so the bottom row of figures essentially represents the logical negation of the top row.



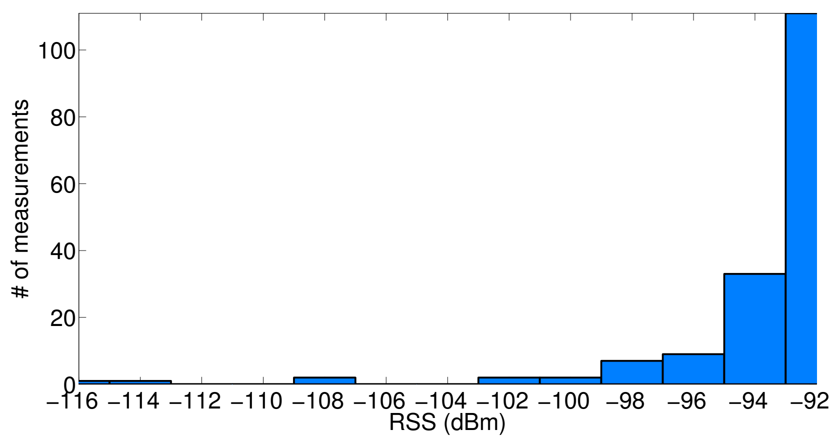
(a) TBM foreground (b) FABS foreground (c) FABS-MMCL-R foreground

(d) TBM background (e) FABS background (f) FABS-MMCL-R background

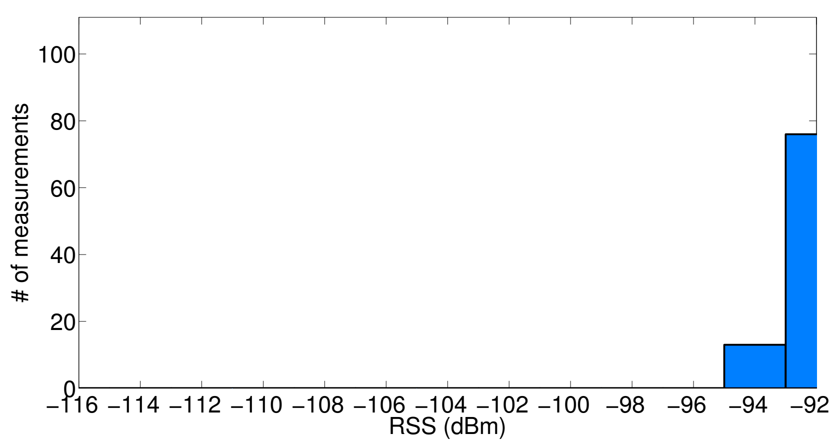
Fig. 5.5 Links assigned to the foreground and background by various background subtraction algorithms for a single frame in experiment set 1 (conducted in an outdoor field). The known position of the obstruction during this frame is marked in blue.

As we described in Section 3.4, FABS is nominally designed to grow the set of foreground links created by TBM. However, in Figure 5.5, we can see that, in this case, FABS is actually removing links from the set created by TBM. At first, we might hypothesize that this is occurring because TBM is being overly-greedy, perhaps due to the θ parameter being set too high. Recall though, that the parameters have all been carefully adjusted in order to optimize performance. Looking at Figure 5.5 just provides more supporting evidence, then, that for outdoor environments, the low-complexity TBM algorithm is already doing such an admirable job of identifying foreground links that any attempt to improve on its efforts using FABS or FABS-MMCL is futile.

Having examined the background vs. foreground behaviour of all the links in the network for a single frame, we will also look at the behaviour of a single link in the network over all frames. Specifically, in Figure 5.6(a), we can see a histogram of the RSS measured on a particular link in the WSN (in this case, the link connecting the nodes located at $(7,1)$ and $(1,0)$) over all frames in K, and in Figure 5.6(b), we can see a histogram of the subset of these values deemed to be in the background by TBM. From these figures, we see that the outliers from Figure 5.6(a) are properly identified as being in the foreground and are removed in Figure 5.6(b). This behaviour is quite representative of the behaviour seen on all the links in the network.



(a) All RSS measurements



(b) Background RSS measurements as determined by TBM

Fig. 5.6 Histogram of RSS measurements seen on the link connecting the nodes located at $(7,1)$ and $(1,0)$ during experiment set 1 (conducted in an outdoor field).

5.3.2 Results for the Indoor Experiment (Experiment Set 2)

Figure 5.7 presents the root mean square estimation error obtained when using our background subtraction algorithms to estimate baseline RSS values for an experiment conducted in an indoor lab. Table 5.3 presents the associated parameter values.

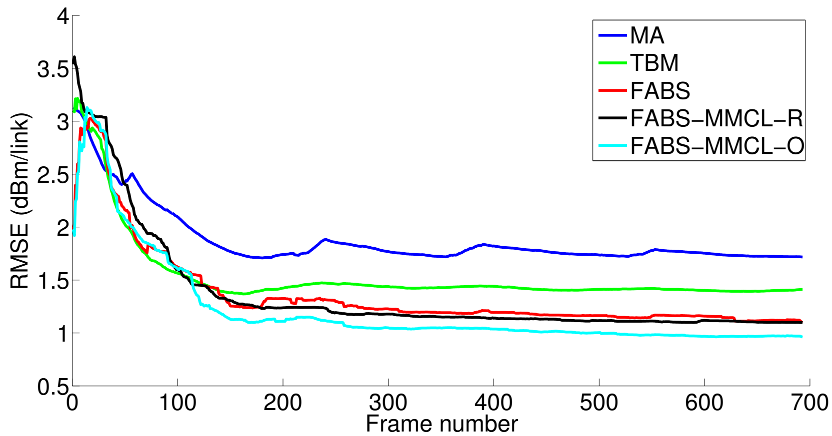


Fig. 5.7 RMSE obtained when estimating the baseline RSS for experiment set 2 (conducted in an indoor lab). The 693 frames used were collected while a walker made four round trips around the path seen in Figure 5.2(b) over a period of ≈ 80 seconds.

Parameter	TBM	FABS	FABS-MMCL-R	FABS-MMCL-O
N	35	35	35	35
σ_t^2	17	17	17	17
θ	0.05	0.05	0.05	0.05
τ	–	0.75	0.75	0.75
σ_s^2	–	10	10	10
η	–	5	5	1
γ	–	–	25	50
J	–	–	35	–

Table 5.3 Optimal background subtraction parameter values for experiment set 2 (conducted in an indoor lab) for various algorithms.

If we compare Figure 5.7 to the similar figure generated for the experiment set conducted outdoors, we can see some important differences. First, if we compare the performance of MA in each case, we see that even this simple approach performs much better in the outdoor environment. This seems to hint that our task of properly estimating the baseline RSS is far harder indoors than it is outdoors. Fortunately, TBM still provides an improvement over MA in the indoor case and, moreover, the more complex FABS and FABS-MMCL provide further improvement still (with FABS-MMCL-O providing more improvement than FABS-MMCL-R). Even after applying these three algorithms, the lowest estimation error which we can achieve is still higher than the estimation error seen for the outdoor experiment sets, but this is understandable given the increased complexity and more intense multipath effects inherent in an indoor environment. Again, additional RSS estimation results which show similar behaviour for additional walks are presented in Appendix A.

We can see another result of these multipath effects if we compare the convergence rates between the indoor and outdoor experiment sets. This convergence rate depends on the amount of noise affecting the RSS measurements and, in particular, on how often we have access to background measurements for each link. At one extreme, if there is absolutely no measurement noise and if no links are ever deemed to be in the foreground, $\hat{\mathbf{R}}_\mathbf{B}$ can converge after a single frame. On the other hand, if many links are in the foreground in each frame (e.g., because there are many obstructions, because the obstruction(s) are very large or because multipath effects cause the obstruction(s) to affect many links) or if the same links are in the foreground for many frames (again, because the obstruction(s) affect many links or because they do not move around very much), it may take several frames before we can obtain a background measurement for each link. The estimate $\hat{\mathbf{R}}_\mathbf{B}$ will then converge more slowly.

In our case, the same size obstruction was moving at approximately the same rate in both the indoor and outdoor sets of experiments. However, due to multipath effects, more links are in the foreground for each frame in the indoor environment (as will be seen when we graphically examine the foreground links selected for a single frame). Therefore, fewer measurements are available for use in calculating $\hat{\mathbf{R}}_\mathbf{B}$ at any given time, causing slower convergence.

Again, we also use Li et al. and Chen's tracking algorithm [8,9] to detect the movements of a person walking around the lab. The tracking errors seen over 100 realizations of the algorithm (initialized with $\tilde{\mathbf{R}}_\mathbf{B}$ and $\hat{\mathbf{R}}_\mathbf{B}$) are presented in Table 5.4.

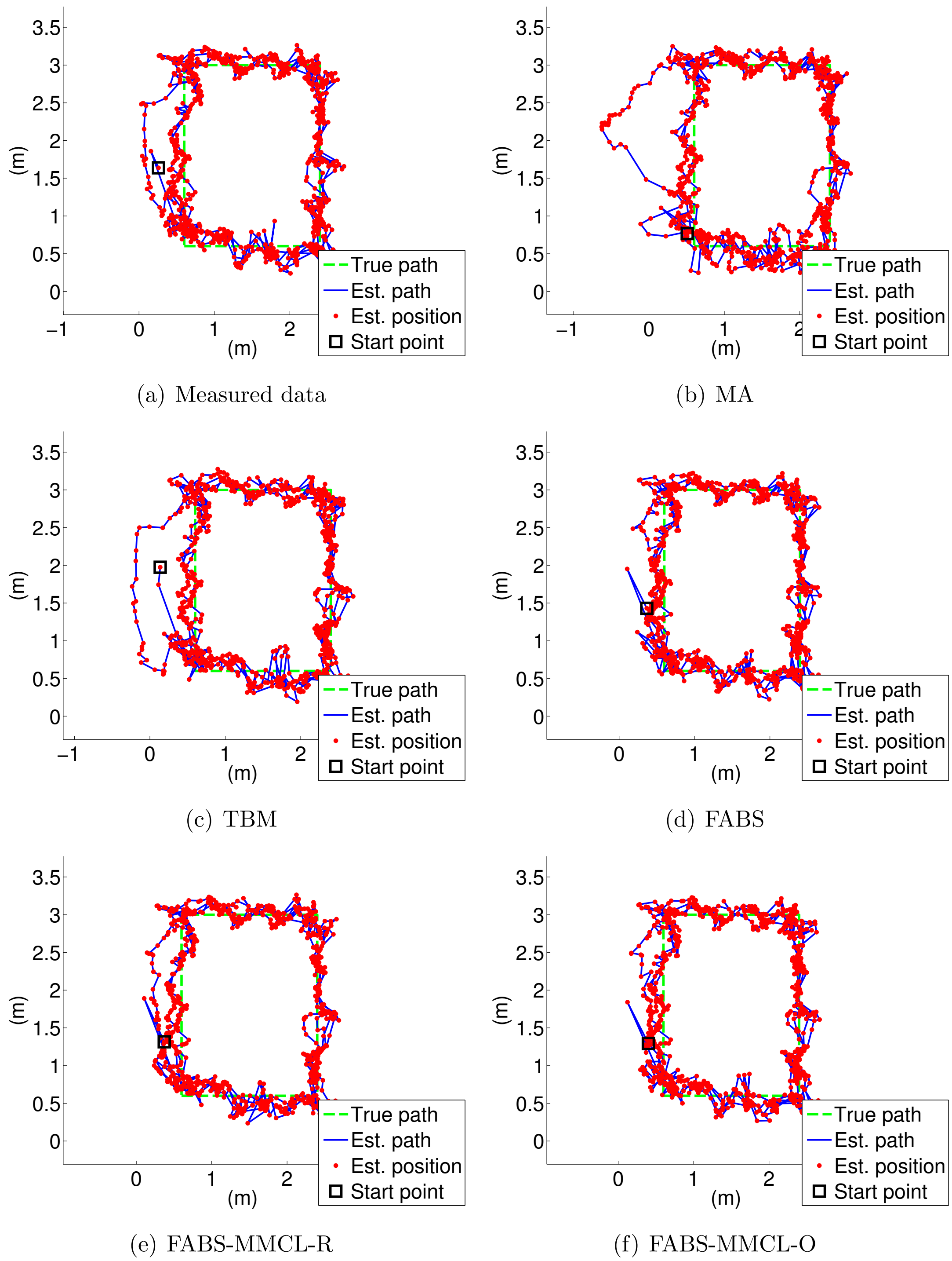
Algorithm	Tracking Error (m)
Calibration Data	0.6060 ± 0.0022
MA	0.6801 ± 0.1763
TBM	0.6404 ± 0.1962
FABS	0.5994 ± 0.0033
FABS-MMCL-R	0.6221 ± 0.0841
FABS-MMCL-O	0.6120 ± 0.0956

Table 5.4 RMSE for tracking using experiment set 2 (conducted in an indoor lab). The tracking algorithm is initialized in turn with measured values of the baseline RSS and with various estimates of the baseline RSS.

As expected, using MA in place of the measured $\tilde{\mathbf{R}}_{\mathbf{B}}$ increases the tracking error from 0.6060 m to 0.6801 m. Simple TBM reduces this error to 0.6404 m while the more complicated FABS-based approaches all reduce the error closer to the level of (or below the level of!) the error seen when $\tilde{\mathbf{R}}_{\mathbf{B}}$ is used. This is presented graphically in Figure 5.8, which plots the average tracks (over 100 realizations) produced when tracking is initialized with different versions of $\hat{\mathbf{R}}_{\mathbf{B}}$. Each point $\bar{\mathbf{z}}_t$ on these paths is equal to

$$\bar{\mathbf{z}}_t = \frac{1}{A}\sum_{a=1}^{A}\hat{\mathbf{z}}_t^{(a)}. \tag{5.5}$$

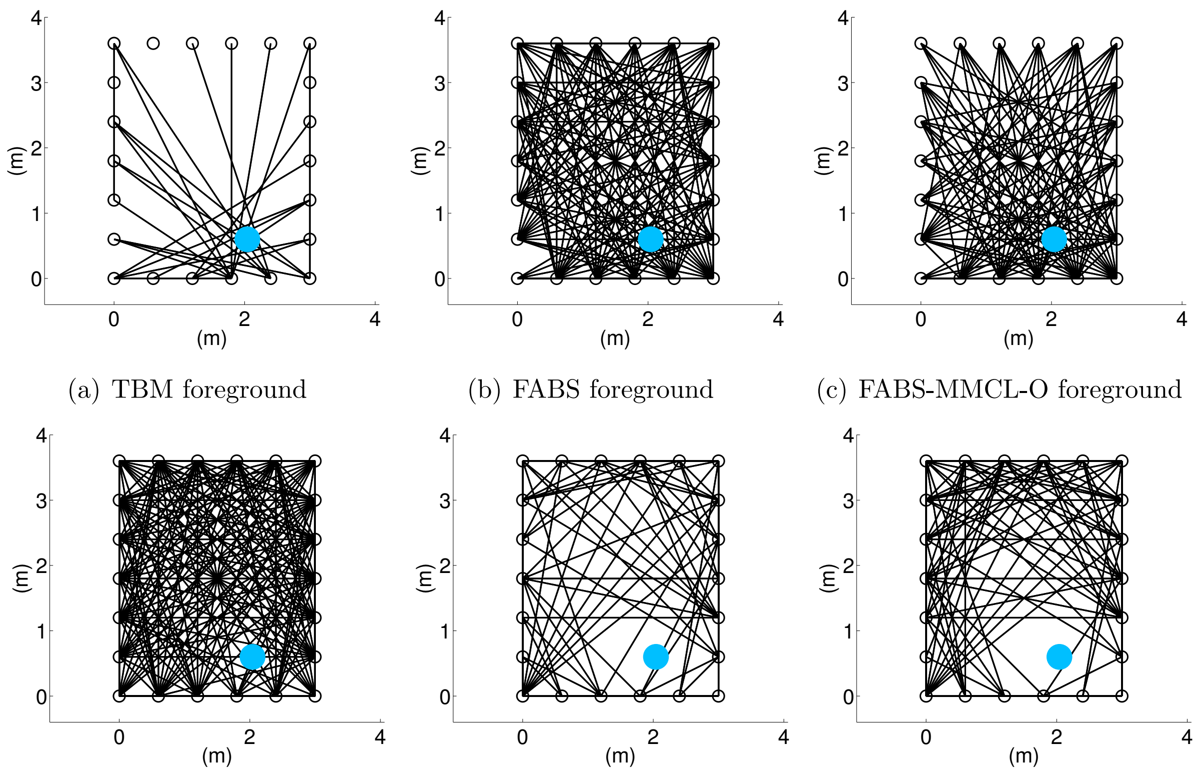
It is particularly interesting to look at the left side of each subfigure, where the estimated path tends to wander away from the ground truth. These data points are the ones produced during the first few time steps of the tracking algorithm. We can therefore conclude that, while all our $\hat{\mathbf{R}}_{\mathbf{B}}$ approximations are sufficiently close to $\tilde{\mathbf{R}}_{\mathbf{B}}$ to allow Li et al. and Chen's algorithm [8, 9] to track our target, the more complex FABS-based methods allow us to find this track more quickly.



(a) Measured data
(b) MA
(c) TBM
(d) FABS
(e) FABS-MMCL-R
(f) FABS-MMCL-O

Fig. 5.8 Tracking results, initialized with $\tilde{\mathbf{R}}_{\mathbf{B}}$ or with $\hat{\mathbf{R}}_{\mathbf{B}}$ as determined by various algorithms, for experiment set 2 (conducted in an indoor lab).

Once again, we also examine which links are assigned to the foreground and background by each algorithm. This is presented in Figure 5.9.



(a) TBM foreground (b) FABS foreground (c) FABS-MMCL-O foreground

(d) TBM background (e) FABS background (f) FABS-MMCL-O background

Fig. 5.9 Links assigned to the foreground and background by various background subtraction algorithms for a single frame in experiment set 2 (conducted in an indoor lab). The known position of the obstruction during this frame is marked in blue.

Unlike with our outdoor experiment set, in this case, we can see that the background subtraction algorithms are behaving as intended. Namely, TBM selects an initial set of foreground links, FABS expands this set, and FABS-MMCL then prunes the set back. Although it may seem like we end up with an unintuitively-large proportion of links being assigned to the foreground in Figure 5.9(c), we can't argue with the fact that large foreground sets such as this provide the closest approximation to the measured baseline RSS. Note, however, that this successful approximation is not just a matter of carelessly assigning a large proportion of the network's links to the foreground. For instance, no matter how we change the parameters of TBM to increase the number of links it assigned to the foreground, we could not decrease the algorithm's estimation error to the level achieved by FABS-MMCL. The finesse of the more complex approaches is required in order to improve the baseline RSS estimate in this indoor environment.

Similar results are seen in the through-wall experiment sets presented in the next section.

5.3.3 Results for the Indoor, Through-Wall Experiment (Experiment Set 3)

Figure 5.10 presents the estimation error obtained when using our background subtraction algorithms to estimate the baseline RSS values for experiments in which measurements were collected indoors, through a solid wall. Table 5.5 presents the associated parameter values.

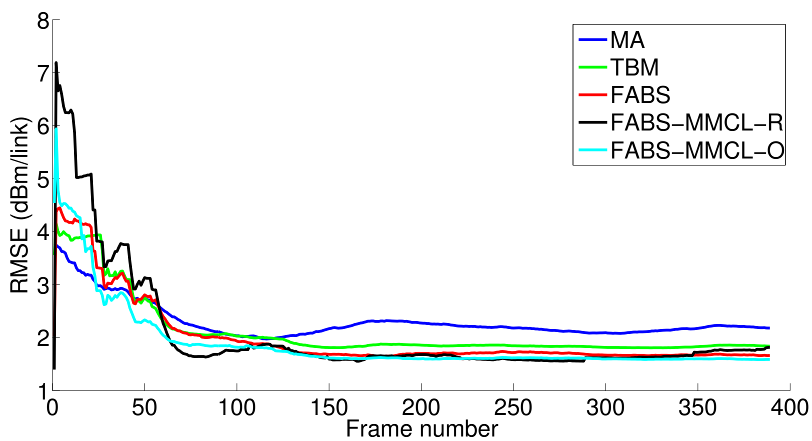


Fig. 5.10 RMSE obtained when estimating the baseline RSS for experiment set 3 (with measurements collected indoors, through a solid wall). The 388 frames used were collected while a walker made slightly less than two round trips around the path seen in Figure 5.3(c) over a period of ≈ 50 seconds.

Parameter	TBM	FABS	FABS-MMCL-R	FABS-MMCL-O
N	35	35	35	35
σ_t^2	1	1	1	1
θ	0.05	0.05	0.05	0.05
τ	–	1	1	0
σ_s^2	–	1	5	10
η	–	5	15	4
γ	–	–	50	10
J	–	–	15	–

Table 5.5 Optimal background subtraction parameter values for experiment set 3 (with measurements collected indoors, through a solid wall) for various algorithms.

In this case, we have once again increased the complexity of our environment, which also increases the lowest achievable estimation error seen in Figure 5.10. Nonetheless, basic TBM still provides an improvement over MA. Likewise, as in the non-through-wall indoor experiment, we can obtain further improvements over TBM by using FABS and FABS-MMCL-O.

Tracking also becomes more complicated when using through-wall measurements. Specifically, it now takes the particle filter a much longer time to converge to the proper track, if indeed it ever does. For the data we collected, whether we initialize tracking with $\hat{\mathbf{R}}_\mathbf{B}$ or with $\tilde{\mathbf{R}}_\mathbf{B}$ itself, the algorithm inevitably requires ≈ 300 time steps before the particle filter can find the correct track. Furthermore, over our 100 realizations initialized with $\tilde{\mathbf{R}}_\mathbf{B}$, in 36 cases, the algorithm never converges at all, resulting in "lost tracks" similar to the one seen in Figure 5.11.

In this environment, it becomes important to consider not just the tracking error for different $\hat{\mathbf{R}}_\mathbf{B}$'s, but also the percentage of lost tracks. This information is summarized in Table 5.6. For instance, when $\hat{\mathbf{R}}_\mathbf{B}$ is calculated using MA, every single one of the 100 realizations results in a lost track. Similarly, when $\hat{\mathbf{R}}_\mathbf{B}$ is calculated using TBM or with FABS, 95% and 93% of the realizations result in lost tracks; consequently, the associated tracking error values are not really reliable. It is only the values of $\hat{\mathbf{R}}_\mathbf{B}$ produced by FABS-MMCL that allow the tracking algorithm to converge in $> 25\%$ of realizations.

Then, in these cases, the tracking error is actually consistently lower than the error seen when the tracking algorithm is initialized using $\tilde{\mathbf{R}}_{\mathbf{B}}$.

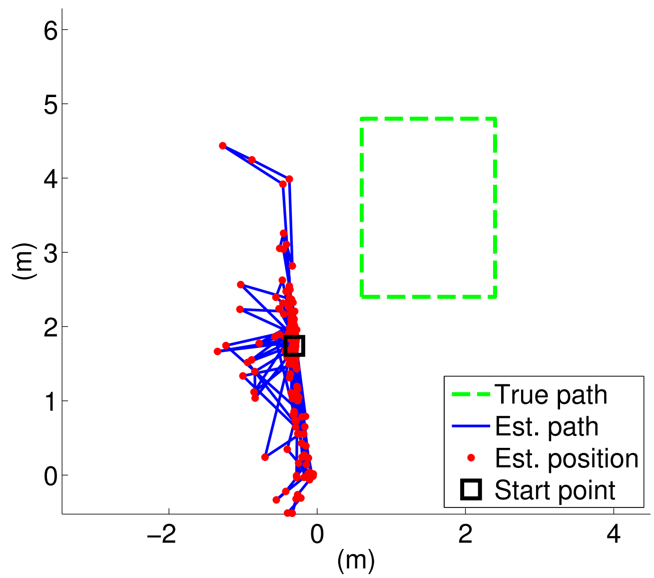
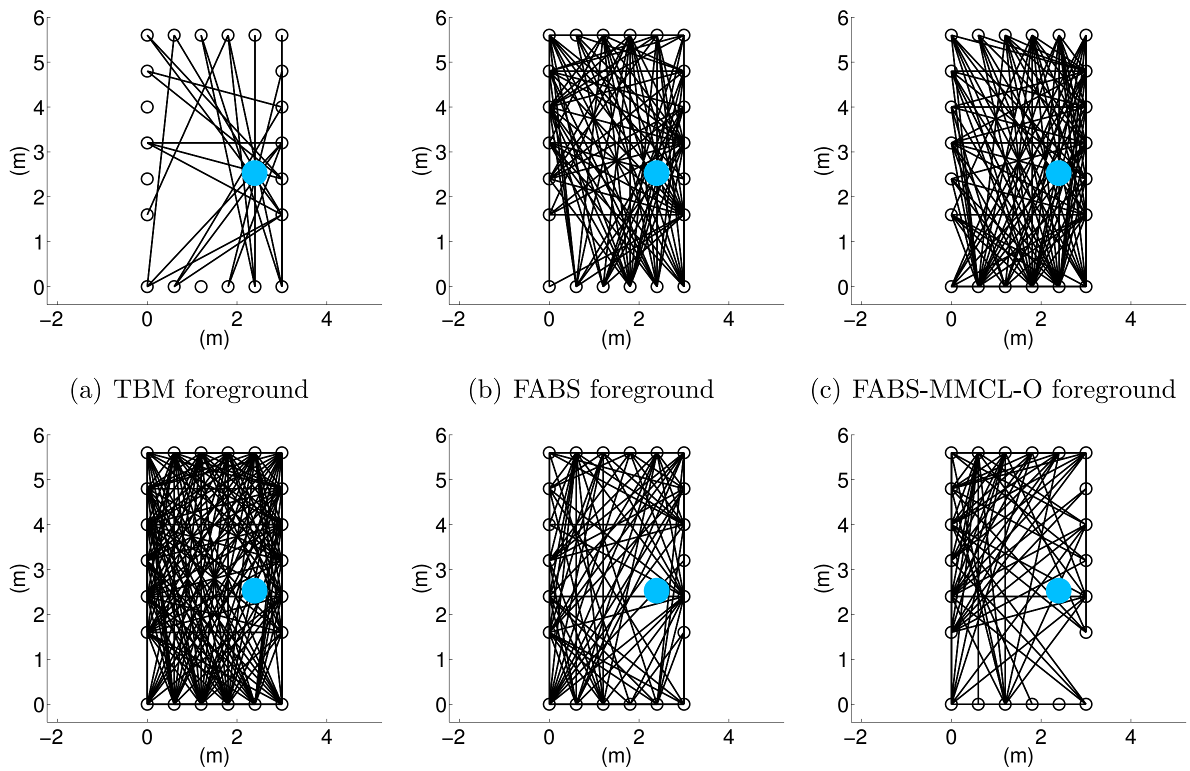


Fig. 5.11 Example of a lost track for experiment set 3 (with measurements collected indoors, through a solid wall).

Algorithm	Lost Track %	Tracking Error (m)
Calibration Data	36	2.7606 ± 0.0116
MA	100	N/A
TBM	95	1.8364 ± 0.8932
FABS	93	2.6290 ± 0.0081
FABS-MMCL-R	74	1.3068 ± 0.2080
FABS-MMCL-O	29	1.7380 ± 0.8578

Table 5.6 Lost track percentage and RMSE for tracking using experiment set 3 (with measurements collected indoors, through a solid wall). The tracking algorithm is initialized in turn with measured values of the baseline RSS and with various estimates of the baseline RSS.

In Figure 5.12, we present a one-frame snapshot of which links in the through-wall environment are being assigned to the foreground and background by each algorithm. Comparing this to the similar figure presented for the indoor environment, we can see that the layout of the foreground is similar in both cases.



(a) TBM foreground (b) FABS foreground (c) FABS-MMCL-O foreground

(d) TBM background (e) FABS background (f) FABS-MMCL-O background

Fig. 5.12 Links selected as being in the foreground and background by various background subtraction algorithms for a single frame in experiment set 3 (with measurements collected indoors, through a solid wall). The position of the obstruction during this frame is marked in blue.

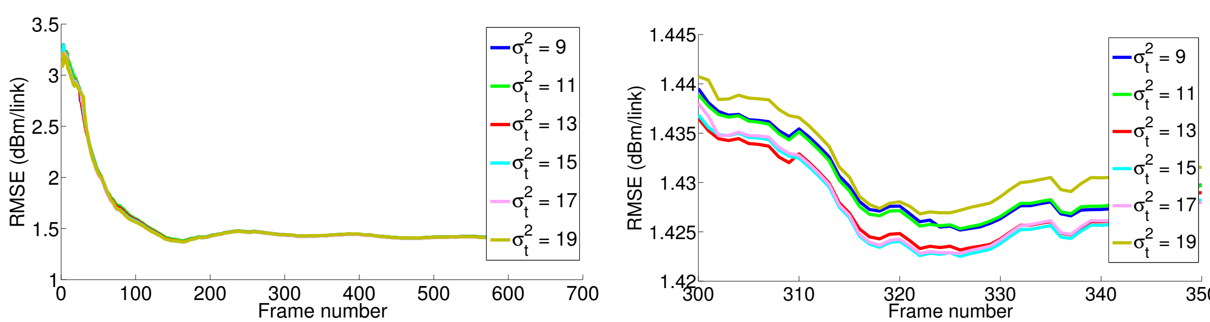
5.4 Sensitivity of the Background Subtraction Algorithms to Variations in the Parameter Values

Thus far, we have obtained our estimation and tracking errors using optimized values for our background subtraction parameters. In practice, depending on how easily these values can be generalized, we may or may not have access to such optimal values for σ_t^2, θ, N, τ, σ_s^2, η, J and γ. Therefore, it is worthwhile to investigate whether our algorithms can still function using sub-optimal values for these parameters.

To do so, we rerun multiple trials of our various background subtraction algorithms on the data obtained during experiment set 2. For each trial, one parameter is varied while all the others are held constant at the ideal outdoor environment values specified in Table 5.3. These results are presented in Figures 5.13–5.20.

These figures provide a glimpse into how sensitive the various background subtraction algorithms are to variations in each parameter, but it is important to keep in mind that this constitutes only a preliminary examination. We are well-aware that the effects of the various parameters cannot truly be isolated from each other. For instance, the ideal value of θ, the decision threshold of the TBM test—and, indeed, the sensitivity of TBM to changes in θ—is highly-dependent on the value of σ_t^2, the variance used in the kernel function associated with TBM. This is because, as σ_t^2 is increased, the variance of the background PDF $P_{\mathcal{B}}$ will likewise increase as per Equation (3.2). This will reduce our sensitivity to θ.

Fortunately, in Figure 5.13, we can see that changing the value of σ_t^2 barely affects our estimation error. This implies that we can choose a fairly large value for this parameter if we wish to decrease our sensitivity to θ.



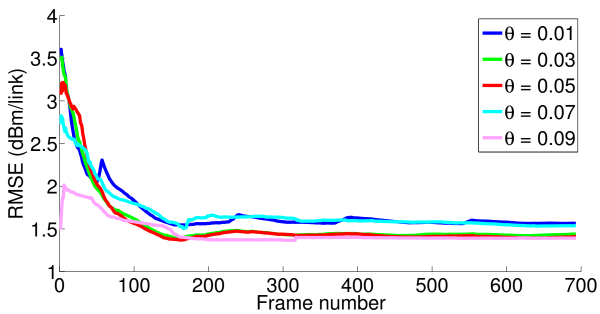
(a) All frames

(b) Detail of frames 300–350

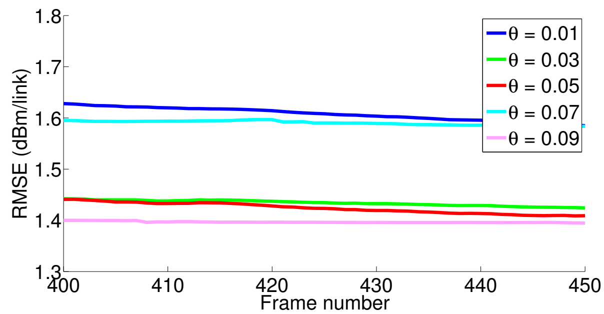
Fig. 5.13 Changes in estimation error when TBM is run using different values of σ_t^2 for experiment set 2 (conducted in an indoor lab).

Even with a well-chosen σ_t^2, we must still be extremely careful when choosing θ: in addition to increasing the RMSE, a poorly-chosen θ can cause another problem as well.

In Figure 5.14, it appears as though the best value for θ is 0.09. Yet, if we consult Table 5.3, which lists the optimal algorithm parameter values, we can see that θ is chosen to be 0.05. This is because, as θ gets larger, more and more links are assigned to the foreground in each frame. For large enough values of θ, some (or all) links will be in the foreground in all frames, and hence we will never have access to any background measurements with which to estimate their baseline RSS. For $\theta = 0.09$, 5% of links are in the foreground in every single of frame. While this may not be a problem for an application where baseline RSS measurements are not required for every link in the network, in our case, we chose to deem the optimal value of θ to be 0.05 rather than 0.09.



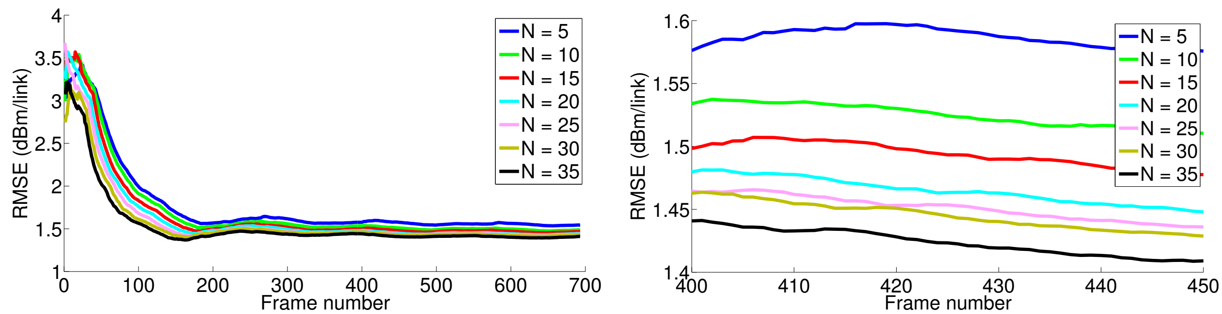
(a) All frames



(b) Detail of frames 400–450

Fig. 5.14 Changes in estimation error when TBM is run using different values of θ for experiment set 2 (conducted in an indoor lab).

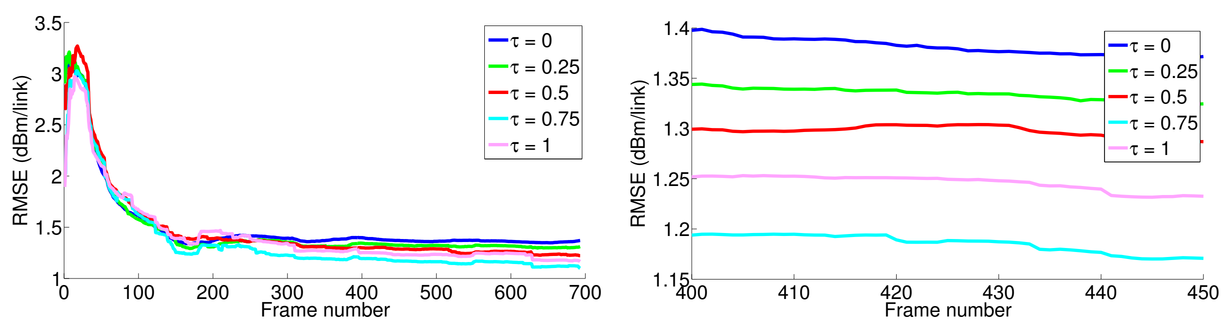
Figure 5.15 presents the estimation error seen for different values of N, the number of frames preceding frame k used to estimate the background in k. We can see that each line in the figure is practically identical except for being shifted slightly down and to the left. Presumably, this behaviour would continue (although not indefinitely) for larger values of N, but, as N increases, the complexity of the background subtraction algorithms increases as well. We make the arbitrary engineering decision to stop increasing N at $N = 35$ and simply state here that someone applying our background subtraction algorithms probably need not be terribly worried about carefully optimizing the value of N.



(a) All frames (b) Detail of frames 400–450

Fig. 5.15 Changes in estimation error when TBM is run using different values of N for experiment set 2 (conducted in an indoor lab).

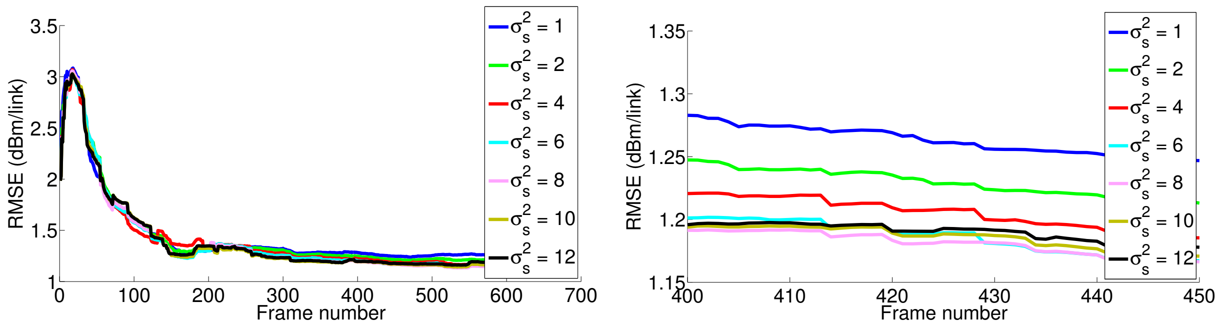
Figure 5.16 presents the estimation error seen for various $\mathcal{L}(\ell)$ neighbourhood sizes as controlled by τ. Recall that $\mathcal{L}(\ell)$ is the neighbourhood used for the FABS algorithm. For this indoor data set, the error decreases as $\tau = 0, 0.25, 0.5$ and 0.75 and then increases again as τ increases to 1. However, the ideal value for this variable is likely heavily tied to the positions of the nodes in the WSN. If the nodes in a network are all separated by 10 m, setting $\tau = 1$ m will have a vastly different effect than if the nodes in a network are separated by 1 m. The results seen in Figure 5.16 are therefore very preliminary. In order to truly determine the effect of changes in τ, we would need to carry out more experiments with many different node deployments.



(a) All frames (b) Detail of frames 400–450

Fig. 5.16 Changes in estimation error when FABS is run using different values of τ to construct $\mathcal{L}(\ell)$ for experiment set 2 (conducted in an indoor lab).

The relationship between σ_s^2 (the variance of the kernel function associated with FABS) and η (the decision threshold for the FABS test) is similar to the one we already described between σ_t^2 and θ. Consequently, we expect the estimation error to be relatively robust against changes in σ_s^2, which is what we see in Figure 5.17.

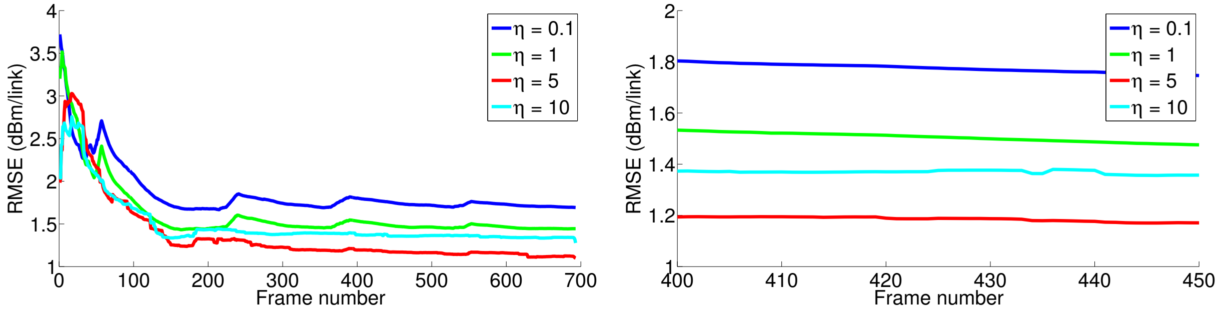


(a) All frames

(b) Detail of frames 400–450

Fig. 5.17 Changes in estimation error when FABS is run using different values of σ_s^2 for experiment set 2 (conducted in an indoor lab).

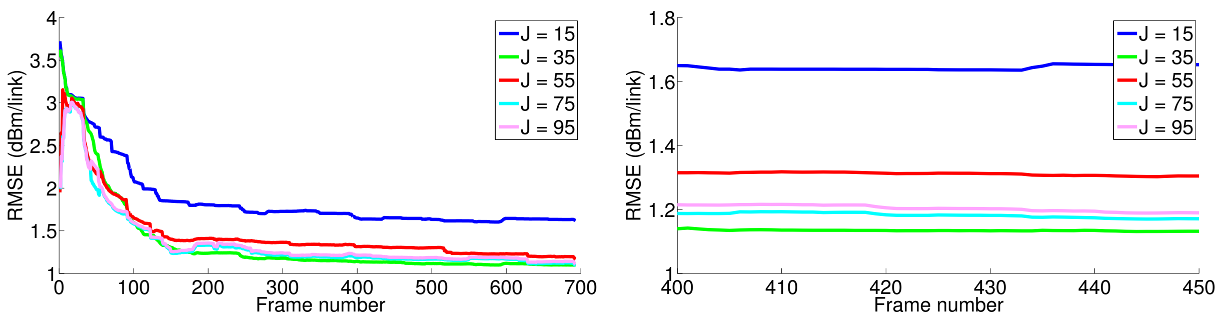
On the other hand, of all the parameters under consideration, the estimation error seems to be most sensitive to changes in η. Although this effect can be mitigated somewhat by choosing larger values of σ_s^2, even for relatively large $\sigma_s^2 = 10$, picking the wrong value of η can increase the estimation error by 33%. This seems worrisome until we remember that FABS-MMCL is actually specifically designed to address this problem by effectively modulating η by making the right-hand side of the FABS-MMCL test in Equation (4.8) larger or smaller.



(a) All frames (b) Detail of frames 400–450

Fig. 5.18 Changes in estimation error when FABS is run using different values of η for experiment set 2 (conducted in an indoor lab).

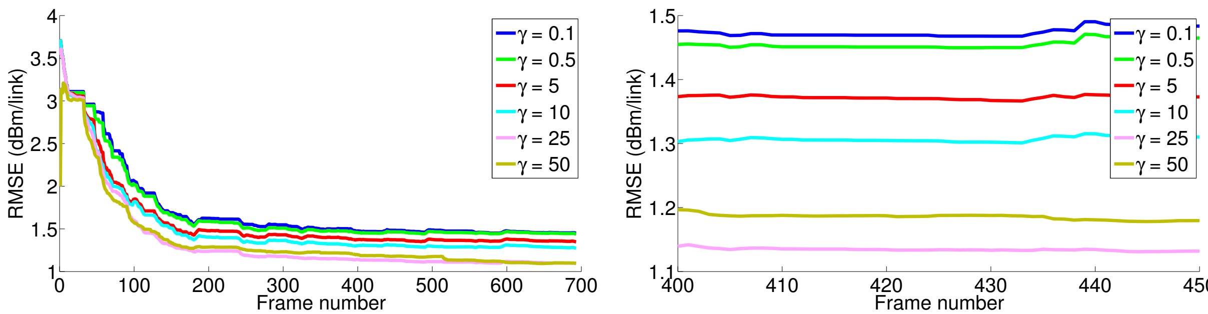
We can now briefly discuss the effects of varying the size of the FABS-MMCL-R neighbourhood $\mathcal{S}(\ell)$ by adjusting the percentage of rectangles J which must be common between two links in order to declare that they are neighbours. However, as with τ, the ideal value for this variable will likely be heavily tied to the number and positions of the nodes in the WSN. Although we can see the estimation error for different values of this parameter in Figure 5.19, and although we can see that our algorithm appears to be fairly sensitive to this parameter, this sensitivity will increase or decrease as the density of nodes in the network changes. Again, the true effects of this parameter cannot be determined until we carry out more experiments using a wide range of node deployments.



(a) All frames (b) Detail of frames 400–450

Fig. 5.19 Changes in estimation error when FABS-MMCL-R is run using different values of J to construct $\mathcal{S}(\ell)$ for experiment set 2 (conducted in an indoor lab).

Finally, Figure 5.20 presents the effects of changing γ. Recall that γ (the natural temperature of the Gibbs distributions of the prior background and foreground probabilities $\pi_{\mathcal{B}}$ and $\pi_{\mathcal{F}}$) influences the modulation of η. Since we already know that our algorithms are highly sensitive to changes in η, it makes sense that they would also be sensitive to γ—although, since γ is used as an exponent, it is actually reassuring that varying it does not have more of an effect compared to what we see in Figure 5.20.



(a) All frames

(b) Detail of frames 400–450

Fig. 5.20 Changes in estimation error when FABS-MMCL-R is run using different values of γ for experiment set 2 (conducted in an indoor lab).

5.5 Algorithm Performance

As discussed in Chapter 3, our various background subtraction algorithms differ substantially in terms of the amount of computation they each require, with TBM being the least computationally-complex, followed by FABS and then by FABS-MMCL. In this section, we examine the calculations required to analyze a single frame of measurements using each algorithm. We then look at how quickly our algorithms perform—in practice—on our various datasets. All the times presented in this section represent the average performance for a single frame seen over 50 repetitions run on a dual-core Intel processor running at 2.13 GHz with 2 GB of RAM.

To start, TBM involves evaluating a Gaussian function N times for each link $\ell \in \Lambda$. Recall that N represents the number of recent frames used to model the background of the frame currently under consideration. Once this is done, we must also run the TBM test seen in Equation (4.2) on each link. This leads to a complexity of

$$O(N\Lambda) + O(\Lambda) = O(N\Lambda). \tag{5.6}$$

In practice, this entire analysis is quite fast, as can be seen in Table 5.7

Experiment Set	Mean Time/Frame (ms)
Set 1 ($N = 25, \Lambda = 231$)	1.0560
Set 2 ($N = 35, \Lambda = 231$)	1.3952
Set 3 ($N = 35, \Lambda = 231$)	1.2780

Table 5.7 The per-frame run times observed for the TBM algorithm. The indicated parameter sets represent those seen in Tables 5.1, 5.3 and 5.5.

Next, in order to run FABS:

1. TBM must be run first to obtain preliminary background and foreground labels for each pixel and to build the background model.
2. Then, $\left|\mathcal{L}^{(k)}_{\mathcal{F}_0}(\ell)\right|$ Gaussians must be evaluated for each link. Here, recall that $\left|\mathcal{L}^{(k)}_{\mathcal{F}_0}(\ell)\right|$ refers to the number of foreground neighbours which exist for the given link as determined by the preliminary test. This number will be different for each link and for each frame; in our experiments, $\left|\mathcal{L}^{(k)}_{\mathcal{F}_0}(\ell)\right|$ ranges from 2 to over 200.
3. Next, the FABS test seen in Equation (4.5) must be evaluated for each link.
4. Finally, steps 2–3 must be repeated a small number of times in order for the labels to stabilize.

If we refer to the number of repetitions needed as ω_F, this results in an overall complexity for FABS of

$$\underbrace{O(N\Lambda) + O(\Lambda)}_{\text{TBM}} + \underbrace{O\left(\omega_F \left|\mathcal{L}^{(k)}_{\mathcal{F}_0}(\ell)\right| \Lambda\right) + O(\omega_F\Lambda)}_{\text{FABS}}. \tag{5.7}$$

The first two terms in this expression represent the initial TBM analysis (step 1) while the latter two terms represent the analysis which is unique to FABS. This expression may be limited by either $O(N\Lambda)$ or by $O\left(\omega_F \left|\mathcal{L}_{\mathcal{F}_0}^{(k)}(\ell)\right| \Lambda\right)$, depending on the values chosen for the algorithm parameters and on the geometry of the network.

The times required to carry out the "uniquely-FABS" part (steps 2–4) of the FABS algorithm can be seen in Table 5.8. The minimum, maximum and mean run times per frame are all shown, to illustrate the possible—although slight—effects of varying neighbourhood size. To determine the full run time of FABS, the times indicated in this table should be added to those required to carry out step 1, although the run time of this step is negligible in practice.

Experiment Set	Minimum Time/Frame (ms)	Maximum Time/Frame (ms)	Mean Time/Frame (ms)
Set 1 ($\tau = 0, \sigma_s^2 = 4, \eta = 1$)	40.7283	46.9627	42.3001
Set 2 ($\tau = 0.75, \sigma_s^2 = 10, \eta = 5$)	21.7503	29.0616	22.3163
Set 2 ($\tau = 0.75, \sigma_s^2 = 10, \eta = 1$)	22.2152	25.1664	22.4699
Set 3 ($\tau = 1, \sigma_s^2 = 1, \eta = 5$)	60.7913	65.2499	61.4724
Set 3 ($\tau = 1, \sigma_s^2 = 5, \eta = 15$)	65.9856	71.1632	66.6618
Set 3 ($\tau = 0, \sigma_s^2 = 10, \eta = 4$)	66.7289	74.3292	67.3660

Table 5.8 The per-frame run times observed for the "uniquely-FABS" part (steps 2–4) of the FABS algorithm. The indicated parameter sets represent those seen in Tables 5.1, 5.3 and 5.5. Note that while τ, σ_s^2 and η are not explicitly mentioned in Equation (5.7), all three parameters will indirectly affect $\left|\mathcal{L}_{\mathcal{F}_0}^{(k)}(\ell)\right|$.

Next, in order to run FABS-MMCL-R:

1. TBM and FABS must be run first to obtain preliminary background and foreground labels for each pixel and to build the background and foreground models.

2. Then, the FABS-MMCL test seen in Equation (4.8) must be evaluated sequentially for each link.

3. Finally, step 2 must be repeated several times to optimize the final labels.

If we refer to the number of repetitions needed for any type of FABS-MMCL implementation as ω_{FM}, this results in an overall complexity for FABS-MMCL-R of

$$\underbrace{O(N\Lambda) + O(\Lambda)}_{\text{TBM}} + \underbrace{O\left(\omega_F \left|\mathcal{L}^{(k)}_{\mathcal{F}_0}(\ell)\right| \Lambda\right) + O(\omega_F \Lambda)}_{\text{FABS}} + \underbrace{O(\omega_{FM}\Lambda)}_{\text{FABS-MMCL-R}} . \tag{5.8}$$

Again, the first four terms in this expression represent the initial TBM and FABS analysis (step 1) while the final term represents the analysis which is unique to FABS-MMCL-R.

The times required to carry out steps 2–3 can be seen in Table 5.9. From this, we can see that FABS-MMCL-R takes quite a bit longer to run than FABS. This is largely due to the fact that the FABS test can be run on all links in parallel while the FABS-MMCL-R test must be run on each link sequentially, an operation which is much slower in practice. To determine the full run time of FABS-MMCL-R, the values in this table must be added to the times required to carry out TBM and FABS.

Experiment Set	Mean Time/Frame (ms)
Set 1	196.7983
Set 2	136.3869
Set 3	347.8364

Table 5.9 The per-frame run times observed for the "uniquely-FABS-MMCL-R" part (steps 2–3) of the FABS-MMCL-R algorithm. The parameter sets used are the same as those seen in Tables 5.1, 5.3 and 5.5.

Finally, the complexity of FABS-MMCL-O is similar to that of FABS-MMCL-R, with one difference. For FABS-MMCL-R (and for FABS itself, for that matter), the link neighbourhoods $\mathcal{S}^{(k)}(\ell)$ (and $\mathcal{L}^{(k)}(\ell)$) can be precomputed and easily filtered into $\mathcal{S}^{(k)}_{\mathcal{F}}(\ell)$ and $\mathcal{S}^{(k)}_{\mathcal{B}}(\ell)$ (or $\mathcal{L}^{(k)}_{\mathcal{F}}(\ell)$ and $\mathcal{L}^{(k)}_{\mathcal{B}}(\ell)$) at run-time. On the other hand the link neighbourhoods required for FABS-MMCL-O must be computed on the fly, at a complexity of $O(|\rho(\ell)|\,\Lambda)$ where $\rho(\ell)$ is the set of rectangles through which link $\ell \in \Lambda$ passes. This is because $r \in \rho(\ell)$—the rectangle with the largest number of foreground links in $\rho(\ell)$—cannot be found ahead of time. This results in a complexity of

$$\underbrace{O(N\Lambda) + O(\Lambda)}_{\text{TBM}} + \underbrace{O\left(\omega_F \left|\mathcal{L}^{(k)}_{\mathcal{F}_0}(\ell)\right| \Lambda\right) + O(\omega_F \Lambda)}_{\text{FABS}} + \underbrace{O(\omega_{FM}\Lambda) + O(|\rho(\ell)|\,\Lambda)}_{\text{FABS-MMCL-O}} . \tag{5.9}$$

In practice, this additional term is negligible, leading to the FABS-MMCL-O run times seen in Table 5.10.

Experiment Set	Mean Time/Frame (ms)
Set 2	409.1133
Set 3	373.2039

Table 5.10 The per-frame run times observed for the “uniquely-FABS-MMCL-O” part of the FABS-MMCL-O algorithm. The parameter sets used are the same as those seen in Tables 5.3 and 5.5.

5.6 Summary of Results Over All Experiment Sets

Looking at our results over all our experiment sets, we can see that TBM is at least a moderately effective way to estimate the baseline RSS in all three of the studied environments. As this is the least computationally-complex algorithm—in addition to being the only algorithm suited for certain applications such as node localization—this is very encouraging. We also see that in outdoor environments, TBM is actually the most effective of all our proposed algorithms and that it is capable of estimating the baseline RSS quite accurately. On the other hand, in indoor environments, more complex approaches like FABS and FABS-MMCL can be used to further refine our estimates.

In all three environments, we see that the estimates returned by our algorithms are accurate enough to use for practical applications such as RF tomographic tracking. We can also see that, when using FABS-MMCL, FABS-MMCL-O seems to be preferable to FABS-MMCL-R, both for RSS estimation and for use with tomographic tracking: although the through-wall tracking performance for FABS-MMCL-R is better than that of FABS-MMCL-O, the lost track percentage is far smaller for the latter method.

Additionally, we can obtain a rough estimate of the length of the convergence period required by our methods. This convergence period will vary depending on precisely what is occurring inside the region of interest, but in all our experiments, it takes less than 300 frames (≈ 40 seconds' worth of measurements) for our estimates to converge. Waiting 40 seconds collect enough measurements to obtain estimates for the baseline RSS represents a huge improvement over having to wait for the entire network to be evacuated in order to obtain these values (if evacuating the network is even possible at all).

Furthermore, from our preliminary experiments with different parameter values, we can begin to determine the correct ranges for our algorithm parameters and gain some insight into how sensitive background subtraction for WSNs is with respect to each parameter.

Finally, by analyzing the complexity and run times of our algorithms, we can see that—although FABS-MMCL is the slowest of the three algorithms, as expected—none of our algorithms takes a prohibitively long time to run.

Chapter 6

Conclusion

6.1 Summary

In this thesis, we have explained how several pre-existing background subtraction algorithms can be adapted for use in determining the background, baseline RSS values of links in a wireless sensor network while the network is already online and while obstructions may already be present in the region of interest. This constitutes the first such method proposed to estimate these values in situations when this baseline cannot be measured directly during an offline calibration period. Using experimental data, we have demonstrated that background subtraction techniques can successfully estimate the baseline RSS in a range of different environments. We have also shown that these estimates are close enough to measured values of baseline RSS to be useful in real-world RF tomography applications such as mean-based RF tomographic tracking.

In Chapter 2 of this thesis, we described RF sensing networks. We explained how they are constructed, and we gave an overview of the relevant physics which dictate how RSS behaves at various distances, in the presence of obstructions and in multipath environments. We then explained the theory underlying four RF sensing network applications: RF tomographic imaging, mean-based RF tomographic tracking, variance-based RF tomographic tracking and RSS-based node localization. This information served both to motivate the search for an accurate estimate of the baseline RSS and to illustrate some of the difficulties faced in obtaining this estimate.

In Chapter 3, we described background subtraction, a technique which originated in the fields of computer vision and image processing for use in determining when pixels in a series of video frames are depicting part of the static background image and when they are depicting the time-varying foreground. We gave a high-level overview of background subtraction and then described three particular background subtraction algorithms. The first of these, background subtraction with temporal background modelling, relied on an assumption of temporal similarity over multiple frames in order to build a model of the background image. Next, we introduced foreground-adaptive background subtraction which used an assumption of spatial similarity across several pixels in the same frame to build a model of the foreground image. Finally, we introduced foreground adaptive background subtraction with Markov modelling of change labels, which used an assumption of spatial ergodicity to model the prior probabilities of a pixel being in the foreground or the background.

In Chapter 4, we brought background subtraction and RF sensing networks together. We pointed out the parallels between the problems of performing background subtraction and estimating baseline RSS, and we explained the similarities between pixel intensities in frames of video and measurements of RSS on links in a wireless sensor network. Then we described how to adapt the assumptions of similarity and ergodicity underlying our three chosen background subtraction algorithms to the WSN domain. In doing so, we proposed transforming spatial similarity into a metric of similarity-of-length, and we introduced two simple methods for modelling spatial ergodicity in WSNs. The first of these methods involved breaking the region of interest into rectangles and then populating neighbourhoods with links which shared a certain percentage of common rectangles; the second method attempted to predict when a link was located near an obstruction by searching for areas along the link's LOS path which contained a large number of foreground links.

Finally, in Chapter 5, we evaluated our proposed background subtraction algorithms in outdoor, indoor and through-wall environments, using experimental data collected by our collaborators at the Beijing University of Posts and Telecommunications. We saw that, in an indoor environment, the least complex background subtraction algorithm which we proposed—background subtraction with temporal background modelling—was capable of estimating the baseline RSS to within an average error of ≈ 0.23 dBm/link. This represented an improvement of ≈ 0.30 dBm/link over a naive mean approximation. More complex background subtraction algorithms did not provide any improvement in this case. In an indoor environment, we again saw that background subtraction with temporal background modelling could estimate the baseline RSS more accurately than the mean approximation. However, in this environment, we saw the best results with obstruction-based foreground-adaptive background subtraction with Markov modelling of change labels. This algorithm was able to approximate the baseline RSS to within an average error of ≈ 0.97 dBm/link, an improvement of ≈ 0.77 dBm/link over the mean approximation. Finally, in a through-wall environment, we again saw that obstruction-based foreground-adaptive background subtraction with Markov modelling of change labels was best-suited to estimating the background RSS, doing so with an average error of ≈ 1.6 dBm/m, improving over the mean approximation by ≈ 0.58 dBm/m.

In the second half of Chapter 5, we also evaluated the performance of an RF tomographic tracking algorithm which was initialized with estimates of the baseline RSS which we built using background subtraction. We found that—while the naive mean approximation sometimes provided baseline RSS estimates which were completely useless to the tracker—initializing the tracker with estimates found using background subtraction could produce tracking results comparable to those obtained when the tracker was initialized with direct measurements of the baseline RSS.

Lastly, we examined the run times of our various algorithms to reassure ourselves that none of them—not even the most complex FABS-MMCL algorithm—took a prohibitively long time to run.

6.2 Future Work

From these results, we can see that the adapted background subtraction algorithms presented in this thesis are definitely capable of estimating baseline RSS accurately enough for use in real-world applications. Still, there is room for improvement, particularly if the algorithms are to be used in complicated through-wall environments which suffer from significant multipath effects. In the future, we would like to explore more complex ways of modelling the spatial similarity and spatial ergodicity between links, in the hope that this could help us see better performance in indoor environments. For instance, we are currently experimenting with creating weighted neighbourhoods which—instead of simply encoding a binary neighbour/non-neighbour relationship between pairs of links—preserve the idea that some of a link's neighbours may be more similar to it (in length, in physical location, in covariance as predicted by the NeSh model [30], etc.) than others.

We are also looking into extending the algorithms described in this thesis so that they can be used to estimate the baseline variance of the RSS seen on a link. The results of this new algorithm could then be used to initialize variance-based RF tomographic tracking methods such as the one described in Section 2.5. Before background subtraction can be applied to RSS variance values, we will have to re-examine the concepts of temporal similarity, spatial similarity and spatial ergodicity yet again in order to determine, for example, if the RSS variance on links which are physically close together is correlated in some way.

There is also plenty of room for work to be done to further our understanding of how best to optimize the background subtraction parameters (θ, N, etc.). Although we provided a brief study of these parameters, a true understanding of their effects will not be complete without a full experimental survey which looks at the ideal parameter sets for different types of environments; for different types, sizes and numbers of obstructions; for different network layouts, etc.

Finally, it would also be interesting to see whether background subtraction could be incorporated iteratively with an RF tomographic tracking. It seems highly probable that the performance of background subtraction would improve if it had access to some a priori information about the locations of any obstructions in the region of interest. Of course, our initial assumption, was that we had no way to learn this information. Yet it is possible that background subtraction could be used to initialize an RF tomographic tracking algorithm, which could then provide the background subtraction algorithm with estimates of obstruction position. This information could be used to refine the baseline RSS estimates, and these estimates could, in turn, be used to refine tracking performance and so on.

Even in the absence of these improvements and extensions, however, we can still say that using background subtraction to estimate the baseline RSS in already-online wireless sensor network is an interesting new technique which has already yielded promising results and which fills a definite need in the domain of RF sensing network applications.

Appendix A

Additional Experimental Results

In this appendix, we present estimation error data from an additional set of outdoor experiments conducted at the Beijing University of Posts and Telecommunications. We also present additional estimation error data from the indoor experiment set discussed in Section 5.1.

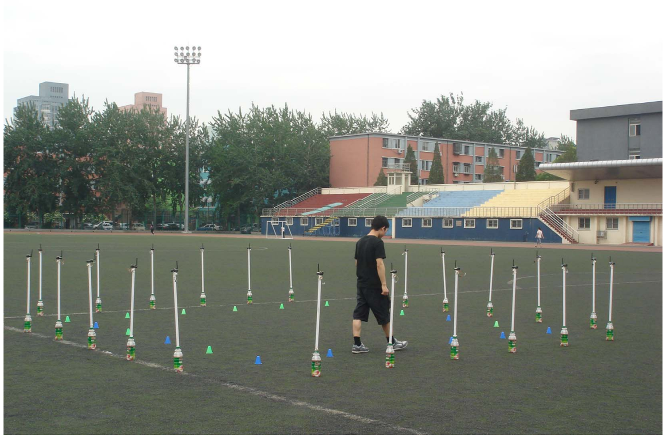
In each case, we present graphs of the estimation error plotted for various background subtraction methods, for several different “walks”. We do not attempt to combine these graphs in any way, e.g., by trying to average over multiple walks to determine the mean error per frame as we do not feel that frames are directly comparable in this fashion. As discussed in Section 5.3.2, each walk will tend to have a different convergence rate related to the exact speed of the obstructions located in the region of interest. For a walk where the obstruction is moving more rapidly and we are able to obtain background measurements for many links over a short period of time, our estimates of the baseline RSS $\mathbf{R_B}$ will converge quickly. On the other hand, if the obstruction lingers in place, preventing us from accessing background measurements for the same links over many successive frames, our estimates of $\mathbf{R_B}$ will converge more slowly. Therefore at, say, frame 100, in some walks we could have measurements for $\mathbf{R_B}$ which are relatively stable, while for other walks, they could still be fluctuating wildly. It makes no sense to average the error of these values together.

If we had access to data from several long walks, this type of analysis could be performed sensibly, but at this time, most of our walk data lasts less than 500 frames.

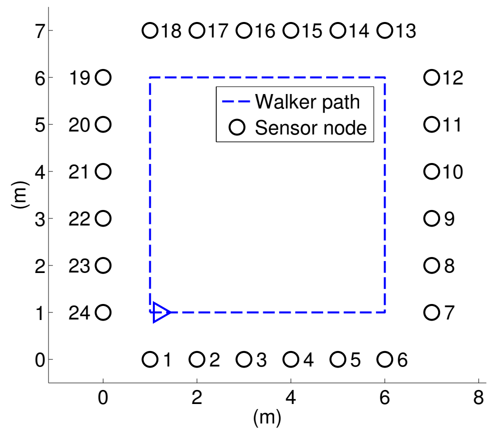
A.1 Additional Outdoor Results

This section provides estimation error data from a set of experiments conducted in a field at the Beijing University of Posts and Telecommunications. This field is different from the one described in Section 5.1.

Figure A.1 shows the node deployment for this set of experiments.



(a) Photo of the experiment setup.



(b) Diagram of the node positions.

Fig. A.1 Node deployment for an additional set of experiments at BUPT, which also took place in an outdoor field.

Figure A.2 shows the root mean square estimation error for a number of walks in an outdoor field at BUPT. Note the differing convergence rates in each figure which arise from how quickly we were able to gain access to background values for each wireless link in the network. Note also that TBM consistently provides the best improvement over the mean approximation, while the more complex FABS-based background subtraction algorithms provide no additional performance boost. The parameter values used to obtain these results are the same as those seen in Table 5.1.

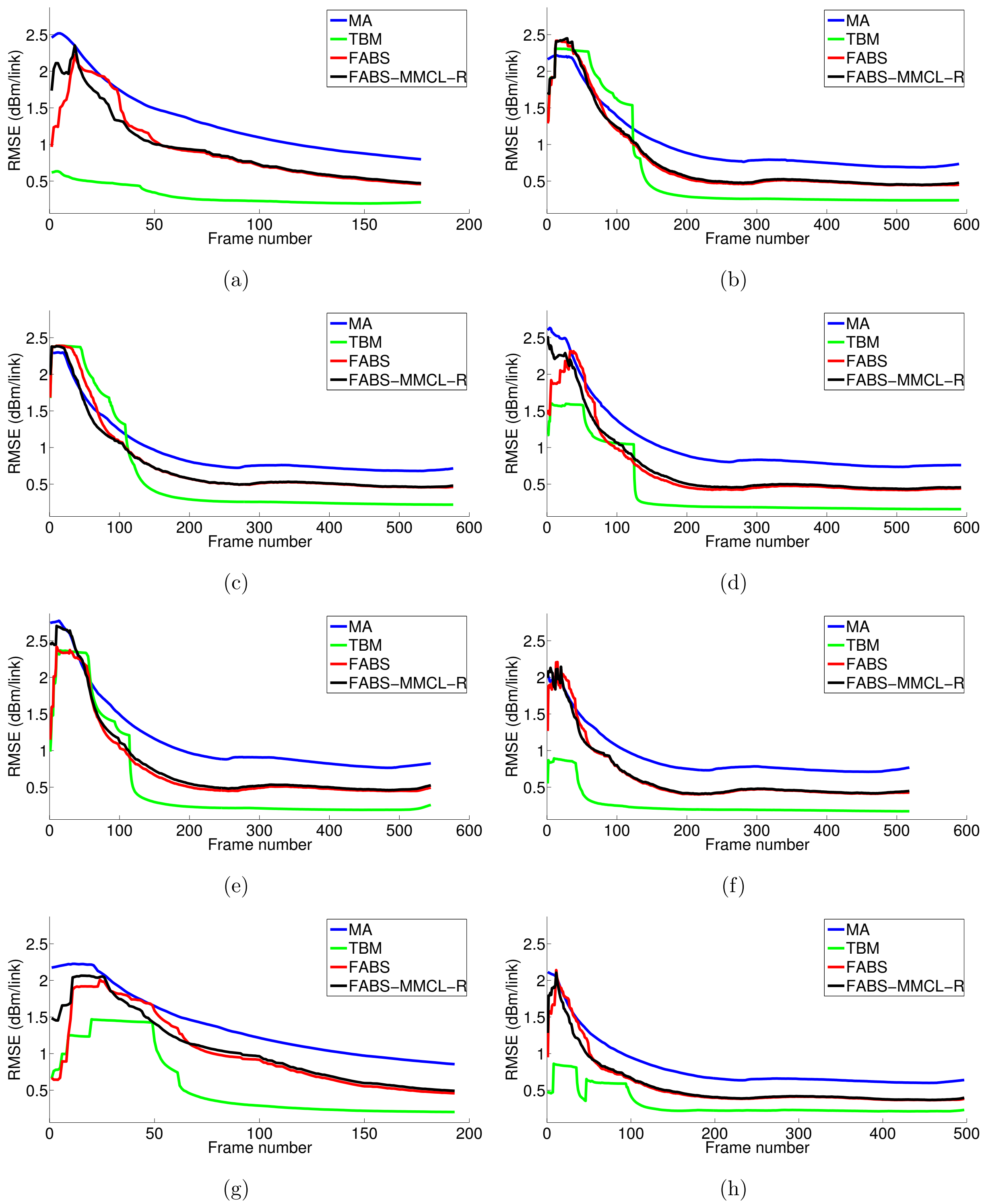


Fig. A.2 RMSE obtained when estimating the baseline RSS for an additional set of experiments (conducted in a second outdoor field), during several different walks.

A.2 Additional Indoor Results

In this section, we present estimation error data from additional walks which are part of the indoor experiment set discussed in Section 5.1. As in the error analysis in Section 5.3.2, we again see that TBM is consistently capable of providing an improvement over the mean approximation while FABS-MMCL-O is consistently capable of improving our estimate still further. The parameter values used to obtain our results are the same as those in Table 5.3.

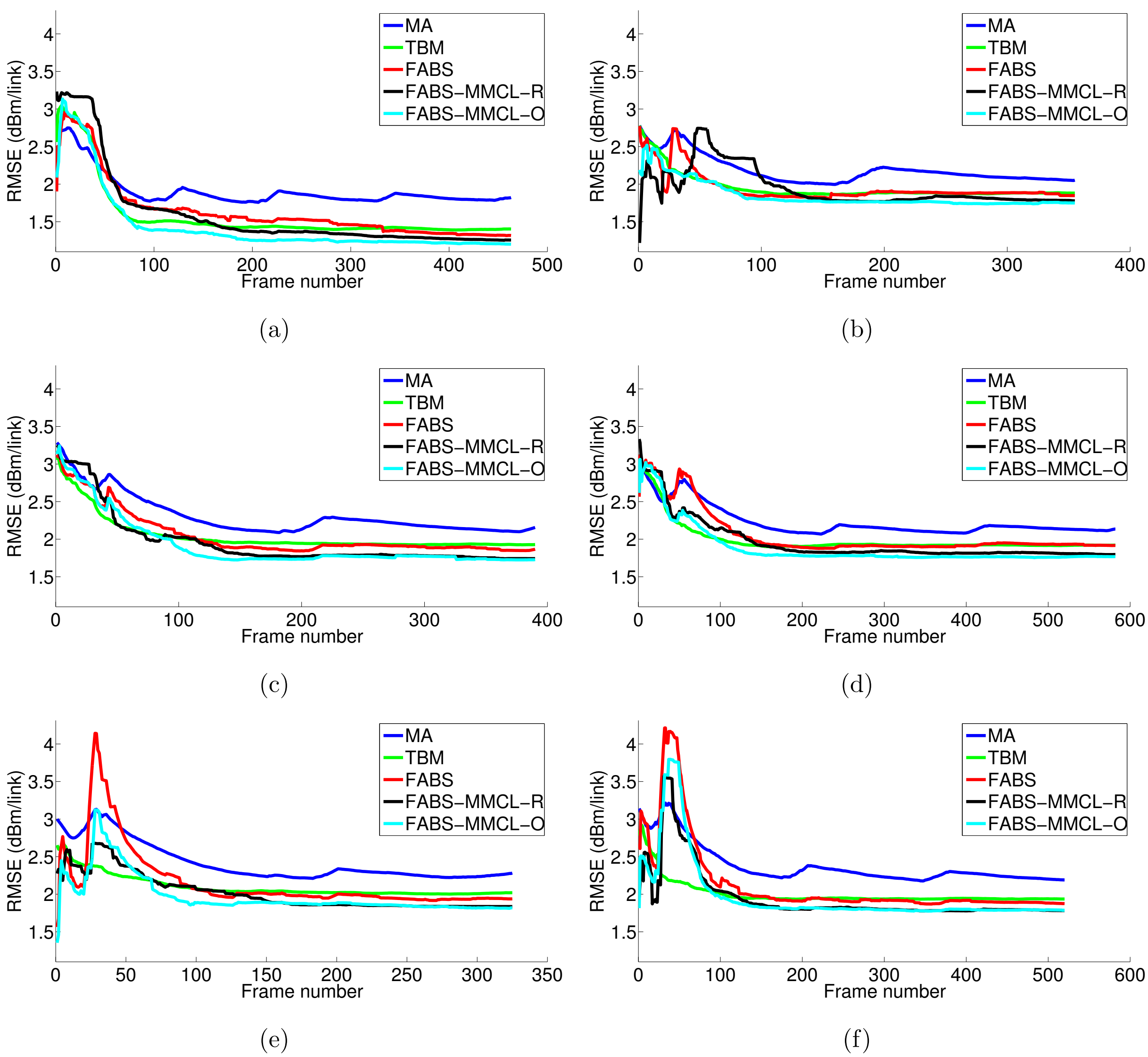


Fig. A.3 RMSE obtained when estimating the baseline RSS for additional walks from experiment set 2 (conducted in an indoor lab).

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[2] B. O'Flyrm, R. Martinez, J. Cleary, C. Slater, F. Regan, D. Diamond, and H. Murphy, "SmartCoast: a wireless sensor network for water quality monitoring," in *32nd IEEE Conf. Local Computer Networks (LCN)*, pp. 815–816, Oct. 2007.

[3] K. Woyach, D. Puccinelli, and M. Haenggi, "Sensorless sensing in wireless networks: implementation and measurements," in *4th Intl. Symp. Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pp. 1–8, Apr. 2006.

[4] J. Wilson and N. Patwari, "Radio tomographic imaging with wireless networks," *IEEE Trans. Mobile Computing*, vol. 9, no. 5, pp. 621–632, May 2010.

[5] M.A. Kanso and M.G. Rabbat, "Compressed RF tomography for wireless sensor networks: centralized and decentralized approaches," in *Proc. 5th IEEE Intl. Conf. Distributed Computing in Sensor Systems (DCOSS)*, Marina del Rey, CA, USA, pp. 173–186, Jun. 2009.

[6] J. Wilson, *Device-Free Localization with Received Signal Strength Measurements in Wireless Networks*, Ph.D. thesis, University of Utah, Salt Lake City, UT, USA, Aug. 2010.

[7] J. Wilson and N. Patwari, "See-through walls: motion tracking using variance-based radio tomography networks," *IEEE Trans. Mobile Computing*, vol. 10, no. 5, pp. 612–621, May 2011.

[8] Y. Li, X. Chen, M.J. Coates, and B. Yen, "Sequential Monte Carlo radio-frequency tomographic tracking," in *IEEE Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3976–3979, May 2011.

[9] X. Chen, "Sequential Monte Carlo radio-frequency tomographic tracking," M.S. thesis, McGill University, Montreal, QC, Canada, May 2011.

[10] X. Chen, A. Edelstein, Y. Li, M. Coates, M. Rabbat, and A. Men, "Sequential Monte Carlo for simultaneous passive device-free tracking and sensor localization using received signal strength measurements," in *10th Intl. Conf. Information Processing in Sensor Networks (IPSN)*, Chicago, IL, USA, pp. 342–353, Apr. 2011.

[11] D. Zhang, J. Ma, Q. Chen, and L.M. Ni, "An RF-based system for tracking transceiver-free objects," in *5th Annual IEEE Intl. Conf. Pervasive Computing and Communications (PerCom)*, pp. 135–144, Mar. 2007.

[12] M. Seifeldin and M. Youssef, "Nuzzer: a large-scale device-free passive localization system for wireless environments," Tech. Rep., Wireless Intelligent Networks Center (WINC), Nile University, Aug. 2009, available at `http://arxiv.org/PS_cache/arxiv/pdf/0908/0908.0893v1.pdf`.

[13] M. Youssef, M. Mah, and A. Agrawala, "Challenges: device-free passive localization for wireless environments," in *Proc. 13th Annual ACM Intl. Conf. Mobile Computing and Networking (MobiCom)*, Montreal, QC, Canada, pp. 222–229, 2007.

[14] N. Patwari, J.N. Ash, S. Kyperountas, A.O. Hero III, R.L. Moses, and N.S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, Jul. 2005.

[15] J.A. Costa, N. Patwari, and A.O. Hero III, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Trans. Sensor Networks*, vol. 2, no. 1, pp. 39–64, Feb. 2006.

[16] L.M. Ni, Y. Liu, Y.C. Lau, and A.P. Patil, "LANDMARC: indoor location sensing using active RFID," in *Proc. 1st IEEE Intl. Conf. Pervasive Computing and Communications (PerCom)*, pp. 407–415, Mar. 2003.

[17] N. Patwari and J. Wilson, "RF sensor networks for device-free localization: measurements, models, and algorithms," *Proc. IEEE*, vol. 98, no. 11, pp. 1961–1973, Nov. 2010.

[18] M. Piccardi, "Background subtraction techniques: a review," in *IEEE Intl. Conf. Systems, Man and Cybernetics*, vol. 4, pp. 3099–3104, Oct. 2004.

[19] T.S. Rappaport, *Wireless Communications Principles and Practice*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 2002.

[20] A.F. Molisch, *Wireless Communications*, 2nd ed. West Sussex, UK: John Wiley & Sons, 2011.

[21] S. Blom, C. Bellettini, A. Sinigalliesi, L. Stabellini, M. Rossi, and G. Mazzini, "Transmission power measurements for wireless sensor nodes and their relationship to the battery level," in *2nd Intl. Symp. Wireless Communication Systems*, Siena, Italy, pp. 342–345, Sept. 2005.

[22] J.M. McHugh, "Probabilistic methods for adaptive background subtraction," M.S. thesis, Boston University, Boston, MA, USA, 2008.

[23] A. Elgammal, R. Duraiswami, D. Harwood, and L.S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proc. IEEE*, vol. 90, no. 7, pp. 1151–1163, Jul. 2002.

[24] J.M. McHugh, J. Konrad, V. Saligrama, and P.-M. Jodoin, "Foreground-adaptive background subtraction," *IEEE Signal Processing Letters*, vol. 16, no. 5, pp. 390–393, May 2009.

[25] J. Besag, "On the statistical analysis of dirty pictures," *Journal of the Royal Statistical Society*, vol. B-48, pp. 259–302, 1986.

[26] S.-C.S. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," in *Electronic Imaging: Visual Communications and Image Processing*, San Jose, CA, USA, pp. 881–892, Jan. 2004, vol. 5308.

[27] M. Moussa and M. Youssef, "Smart devices for smart environments: device-free passive detection in real environments," in *IEEE Intl. Conf. Pervasive Computing and Communications (PerCom)*, pp. 1–6, Mar. 2009.

[28] M. Gudmundson, "Correlation model for shadow fading in mobile radio systems," *Electronics Letters*, vol. 27, no. 23, pp. 2145–2146, Nov. 1991.

[29] Z. Wang, E.K. Tameh, and A. Nix, "Simulating correlated shadowing in mobile multihop relay/ad-hoc networks," Tech. Rep., IEEE 802.16 Broadband Wireless Access Working Group, Jul. 2006, available at `http://www.ieee802.org/16/relay/contrib/C80216j-06_060.pdf`.

[30] N. Patwari and P. Agrawal, "NeSh: a joint shadowing model for links in a multi-hop network," in *IEEE Intl. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2873–2876, Apr. 2008.

[31] P. Agrawal and N. Patwari, "Correlated link shadow fading in multi-hop wireless networks," *IEEE Trans. Wireless Communications*, vol. 8, no. 8, pp. 4024–4036, Aug. 2009.

[32] Texas Instruments, *A True System-on-Chip Solution for 2.4 GHz IEEE 802.15.4 and ZigBee Applications*, 2011, available at `http://focus.ti.com/lit/ds/symlink/cc2530.pdf`.